Personal Software

Issue 1, 1994

Software

Random Data

Personal Software Magazine
IBM®

IBM Personal Software Magazine is published by IBM Personal Software Products User Group Relations, Austin, Texas, U.S.A.

User Group Relations Program Manager
Gene Barlow
Editorial Services MDE Associates
Illustrations Jamison American Originals
Typesetting Dove Oaks Publishing

This publication is distributed only in bulk to PC and OS/2 user groups. Individual copies and subscriptions are not available.

Correspondence should be addressed to Gene Barlow, IBM Corporation, P.O. Box 201449, Austin TX 78720-1449. If you're electronically connected, you can send an Internet note to IBMPCUG@VNET.IBM.COM. IBM employees can send to IBMPCUG at AUSVM1.

IBM believes the statements contained herein are accurate as of the date of publication of this document. However, IBM hereby disclaims all warranties as to materials and workmanship, either expressed or implied, including without limitation any implied warranty of merchantability or fitness for a particular purpose. In no event will IBM be liable to you for any damages, including any lost profits, lost savings or other incidental or consequential damage arising out of the use or inability to use any information provided through this service even if IBM has been advised of the possibility of such damages, or for any claim by any other party.

Some states do not allow the limitation or exclusion of liability for incidental or consequential damages so the above limitation or exclusion may not apply to you.

This publication could contain technical inaccuracies or typographical errors. Also, illustrations contained herein may show prototype equipment. Your system configuration may differ slightly.

IBM has tested the programs contained in this publication. However, IBM does not guarantee that the programs contain no errors.

This information is not intended to be a statement of direction or an assertion of future action. IBM expressly reserves the right to change or withdraw current products that may or may not have the same characteristics or codes listed in this publication. Should IBM modify its products in a way that may affect the information contained in this publication, IBM assumes no obligation whatever to inform any user of the modifications.

Some of the information in this magazine concerns future products, or future releases of products currently commercially available. The description and discussion of IBM's future products, performance, functions, and availability are based upon IBM's current intent and are subject to change.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not imply giving license to these patents.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programming or services in your country.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever.

All specifications are subject to change without notice.

Transforming Business with Object Technology

D'Ann Ostrom, Coordinator
IBM Corporation
Austin, Texas

This article, adapted from a white paper with the same title, outlines IBM's Personal Software Products (PSP) strategy for object technology. PSP will play a leadership role in the development of an object technology environment that will change the economics, and will simplify the process, of software development for our customers. This article looks at industry directions that are driving toward object technology adoption and implementation, benefits to be derived from the technology, current status of what is available, and what is changing in the industry to move object technology into the mainstream. It also includes an overview of PSP object technology product directions, and some suggested steps for getting started with object technology.

Changes in Business Require Changes in Technology

Today's dynamic business environment and key technology innovations are combining to propel object technology (OT) into the mainstream of computing. This convergence comes at a critical time – when corporations are in the midst of a significant global shift in the composition, organization, and operation of their businesses.

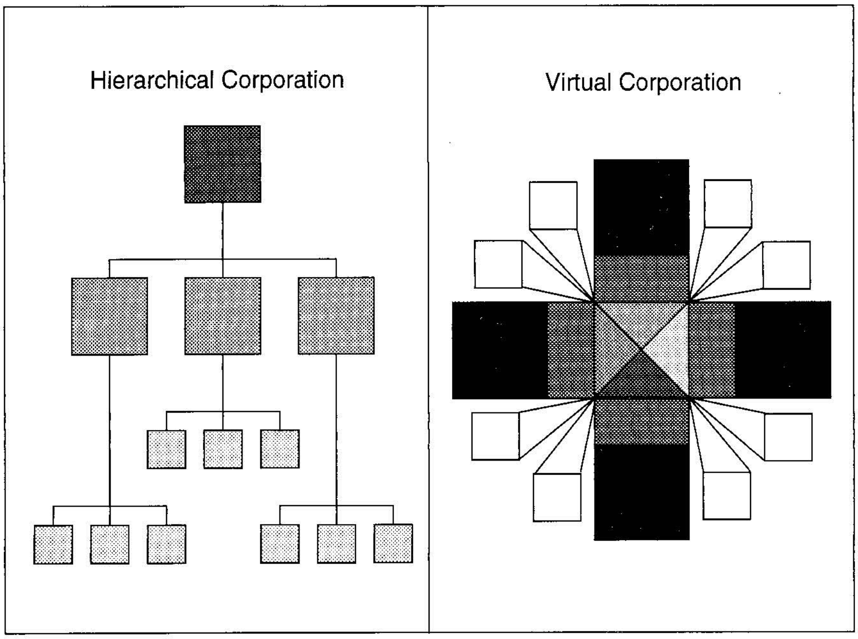


Figure 1. Organizations are Becoming Flat, Networked, and Workflow-Driven

Companies are re-engineering, restructuring, and resizing at an unprecedented rate to become more competitive or, in some cases, to survive. Businesses are transforming themselves to become flat, networked organizations that are process- and workflow-driven.

To deliver viable products and services faster, businesses are exploring new relationships – looking beyond their own organizations to embrace customers, suppliers, and even competitors as a valuable part of the delivery mechanism. The concept can be viewed as a *virtual corporation* – a highly focused organization that emphasizes its core competencies, and relies on long-term alliances and a contracted, dynamic work force tailored to fill specific needs and skills, as shown in Figure 1.

In addition, organizational dynamics and technological innovation are speeding the transition. Computing power that once occupied entire rooms is now on the desktop, in a briefcase, or on the kitchen table at home. This power and availability, coupled with the information needs of a virtual corporation, are forcing a major transition in the information technology (IT) infrastructure.

For most organizations, the days of clerical-based computing – simple, static, centralized information activities – passed long ago. Today, most IT organizations have reached beyond the walls of their enterprises to connect suppliers, as well as their

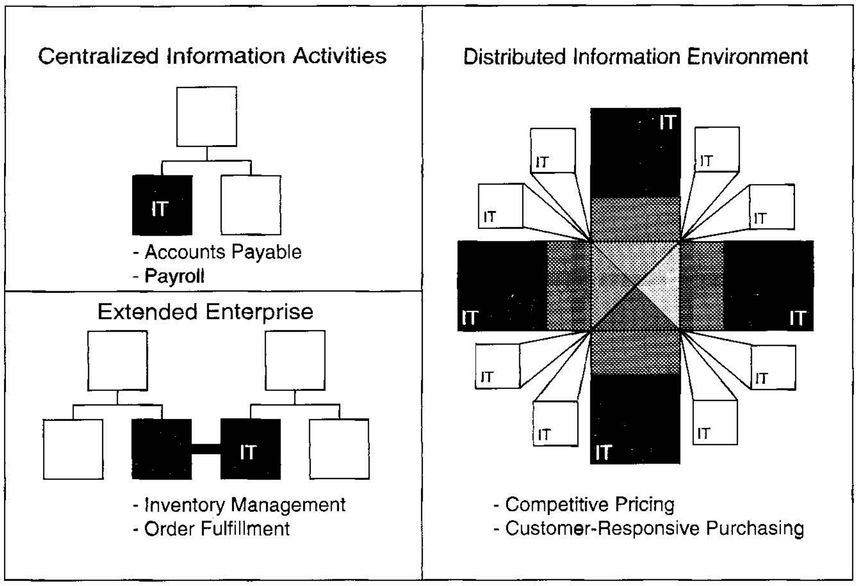


Figure 2. Components of the Distributed Information Environment

fulfillment channels, to smooth product or service delivery, and to compress cycle times. While these will continue to play an important role, the IT infrastructure must also:

- provide real-time access to information that reinforces each user's expertise, and
- adapt easily to the changing business environment – regardless of the people, locations, organizational issues, and processes. (See Figure 2.)

Delivering this infrastructure is an enormous undertaking. To begin with, there is the fundamental complexity of the environment itself, and the incompatibilities of the disparate hardware and software involved. As more and more IT decisions are driven by impatient, profit-motivated, end-user departments, information systems (IS) organizations will face growing pressure to deliver powerful, distributed applications quickly. The ability to involve the people who understand the business problem is critical.

Object technology radically alters the way that software systems are developed, as well as the ways businesses use these systems to achieve competitive advantage. OT changes the economics of application development – and the possibilities.

The promise of OT is compelling: to solve a complex business problem by assembling and/or extending reusable software components. For example, if a bank wants to build an application that makes personal bankers more productive, a developer might start with a loan framework that models the banking business. It might contain a customer information object, a portfolio object, a credit-check object, and a risk-analysis object. Any of these objects could be modified or changed without affecting the rest of the objects in the framework, or the ways in which they interact. So, by altering only the risk-assessment object, a developer could create a specialized loan application for valued customers – with much less effort than if he or she used a current procedures-based approach.

The benefits are evident. Since OT applications more closely model the business problems that they are designed to solve, they are easier to develop and to maintain. Conversations center around loan approval or portfolio, rather than complex, procedure-based syntax. Once written, object programs are reusable and easily extended. Initial programmer effort is used over and over again. Creating a student loan offering, for example, becomes simply a matter of extending the standard loan framework, to adjust the risk analysis object, and to change the amortization period and term of the loan.

However, objects themselves are not enough. They must reside in an open, distributed environment that will allow the use of components in a plug-and-play fashion. This enables the vision of dynamic re-engineering to become a reality. With that vision rapidly approaching, IBM Personal Software Products (PSP) is taking the lead in providing OT-based solutions, frameworks, and services.

The fact is that OT provides the foundation for transforming businesses, and is the catalyst for reshaping the information-processing industry itself. A growing understanding of this potential is leading to fundamental changes in all aspects of the IT industry – services, hardware, software, distribution, even terminology.

Objects: A Solution to Critical Business Problems

In today's competitive environment, businesses of all sizes are recognizing that inflexible and unresponsive systems can no longer sustain their organizations. Although the distributed client/server model has been championed as the way to address

Object Technology Terms and Definitions

New terms are entering the mainstream information technology vocabulary. Here are definitions of the most common terms.

Framework: A set of objects that work together to perform a specific task. Graphical user interface (GUI) frameworks, for example, let objects like windows, menus, and dialogs work together to provide an efficient way to build consistent graphical applications. There are frameworks for business problems too, such as hospital management or inventory control. A well-designed framework provides the general design and implementation for a specific problem, and allows a developer to customize it to a particular situation.

Object: (also called a *component*) The data and logic that represents a useful element in an application. For example, in a financial application, there may be objects that represent account, branch, and customer. The customer object can be defined to perform certain functions, but the details of how it does them are hidden from all but the designer of the customer object itself. Objects are valuable in designing and implementing software because they hide complexity.

Message: The means by which one object requests the services of another. For example, a customer object may send a "print statement" message to a specific checking-account object. The message identifies the method that the object will use to perform the request.

Method: An operation that an object can perform. For example, an account object may have separate methods for performing a debit, a credit, or printing a statement.

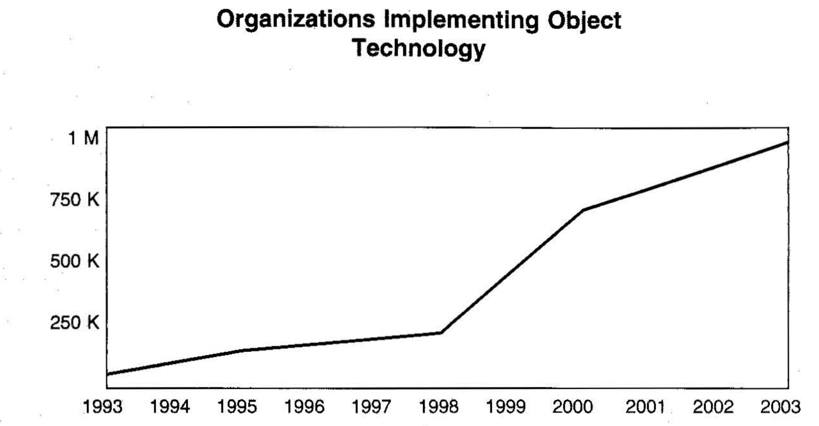
Module: A loosely, yet widely, used term describing a piece of code that performs some function. Objects are sometimes referred to as self-contained modules to emphasize the notion of encapsulation. Usually an object's methods are implemented as code modules.

Object Request Broker (ORB): The mechanism that enables objects to communicate with each other across a network. It provides services for security registration and object management. The Object Management Group (OMG), an industry consortium, has defined the Common Object Request Broker Architecture (CORBA) to specify standards, so that different ORB implementations can work together.

Encapsulation: The practice of making a project self-contained and hiding its internal structure. Thus, an object maintains its own data and the logic to perform operations, while exposing to other objects only what is necessary to request actions. Encapsulation leads to flexible designs, because internal structures – or even whole objects – can be modified without affecting the rest of the system.

Polymorphism: (literally, *many shapes*) The property that allows different objects to respond to the same request. For example, savings-account, checking-account, and loan-account objects may each handle a request to "print statement," even though each statement may contain different information. Polymorphism reinforces the value of encapsulation in an application by reducing the amount of information we need to know about an object. This allows objects to be modified or replaced without affecting other areas of the application.

Inheritance: The mechanism that allows a developer to create a customized object, based on the implementation of a general one. For example, an account object may handle debits and credits, and produce statements. A savings-account object may inherit all of those features and accrue interest on the outstanding balance. Inheritance increases productivity by providing reusable software without sacrificing flexibility.



Source: Workgroup Technologies, Inc.

Figure 3. For Many Companies, the Inevitable Move to OT Has Already Begun

the flatter organizational structure, the complexity involved in implementing these solutions has slowed progress. OT will help accelerate the move to this powerful, multiplatform computing environment by dramatically simplifying the development of distributed applications.

Object technology, by definition, is well suited for distributed systems, because both the business logic and the data are contained within objects, allowing them to be located anywhere within a distributed network. OT can mask or remove the majority of specific platform characteristics. The application development team can then place more emphasis on understanding the marketplace, customer interactions, and business processes. OT also will enable increased user involvement for designing and enhancing applications – and perhaps even for modifying them independently.

With the advent of open, distributed objects, the end user and IT consultant will participate jointly in designing a solution for a set of business problems. First, the business process description becomes the foundation for the solutions. Then, building on an industry framework, the development process can be iterative, resulting in an operational prototype more quickly. Since the solution is expressed in process and simulation terms, the end user can be directly involved in enhancing the solution. The second step begins an iterative process of coding, testing, and redesigning, using additional class libraries of business objects. The third step is maintenance by the end user through enhancement or redesign of the solution on a dynamic basis to meet changing business needs.

As object-oriented (OO) solutions evolve, they will look less like traditional applications, and more like intelligent, interactive business processes. Today, end users experience the procedural world of information processing applications on two basic axes: vertical, mission-critical or industry-specific applications; and horizontal, personal-productivity applications. The user has limited involvement and control of the flexibility and usability of vertical applications, while exercising extensive control over the personal-productivity applications that he or she chooses to use. OT will enable vertical and horizontal applications to evolve to a new generation of diagonal applications – mission-critical, yet heavily influenced and directed by the user's need for flexibility and ease of use.

Market Expectations for Object Adoption

Studies, such as the one in Figure 3, have shown that market and technology forces make the adoption of OT inevitable. Markets are moving toward shorter product life cycles, and are placing recognized value closer to the customer. Although software has made fundamental improvements in function, ease of use, and quality, it has not kept pace with hardware's move toward smaller, cheaper, and faster units. Objects are the software equivalent of the microprocessors that triggered a radical change in the computer hardware industry's business model.

Companies are moving toward OT for two reasons: first, to reduce the time required to develop applications, and second, to design applications that create competitive advantages by improving responsiveness to shifts within their industry. There is growing evidence of a strong market pull for OT from corporations developing complex mission-critical applications. Existing procedural-based technology has reached its limits for many of these firms. In addition, many companies that traditionally have been conservative with their adoption of technology are realizing that they, too, have reached the limits of traditional computing approaches. They are now accepting OT as the way to rebuild their software systems.

A recent International Data Corporation (IDC) study of 800 representative customers reinforced this trend. The study indicates that 12% of surveyed customers are already using OT extensively; 44% are exploring OT benefits; 73% are moving to OT from an existing client/server environment; and 68% are moving to a distributed systems environment while using or exploring OT. All of these customers recognize the potential that OT offers.

A research study by Workgroup Technologies, Inc., shows that a variety of organizations, lines of business within large corporations, companies, and educational institutions have already begun deploying OT applications. And the number of advanced tools becoming available over the next five years will fuel the acceptance of OT.

OT offers the basis for reusable, plug-and-play software that will complement the new infrastructure required to optimize business processes and applications on a dynamic basis. Its principles allow generalized components to be used as is, or more important, to be specialized for a particular implementation. OT enables components to be connected as application and system frameworks to solve complex, mission-critical problems. Despite the potential of OT, a number of critical inhibitors have pre-empted its widespread deployment.

Objects: Why Now?

Given the obvious potential of OT, it is important to understand the problems that the industry has had to overcome, as well as the changes that have taken place, in order to finally make OT a viable option today.

There is growing evidence of a strong market pull for OT from corporations developing complex mission-critical applications.

Problem: *Component incompatibility.*

Despite today's choices of OT languages, components generated by different tools do not work together. Not only is it difficult to use the components outside of a particular domain, it is impossible to customize them without having the source code or operating within the system that created them. For example, if a generalized component is created in C++, it is difficult to use it, and impossible to refine it from Smalltalk or COBOL.

Solution: *Emergence of industry standards.*

To address the issues of compatibility and object standardization, the Object Management Group (OMG) was formed in April 1989. This nonprofit, international corporation is dedicated to establishing industry guidelines and object-management specifications that will provide a common framework for distributed application development.

Standardization will allow components to be packaged in binary form which, in turn, will allow them to be invoked and refined from any supported language, e.g., C++**, Smalltalk**, or Object COBOL. In addition, a logical extension of this cross-language standard is a set of services that will allow components to be located and managed across a network. The ultimate goal is to automate these networks and other functions, so that the developer need only concentrate on application functionality.

The OMG has developed the Object Management Architecture (OMA) as the basis for defining the primary pieces of any object-oriented computing environment. The OMA defines the Object Request Broker (ORB) as the mechanism that sends and receives messages between objects. The ORB is central to distributed OT, because it provides an environment in which different applications running on different computers in different locations and environments can interoperate seamlessly.

The leading standard defined by OMG is the Common Object Request Broker Architecture (CORBA). CORBA is a specification that defines ORB implementations, services, and interfaces. While CORBA is neither perfect nor complete, the OMG now has more than 300 members, and OT vendors are working to be labelled CORBA-compliant. For customers, CORBA compliance provides assurance that the language, tool, or implementation selected provides the minimum amount of acceptable standardization memory necessary to bridge incompatible technologies.

Benefits of Object Technology

Benefits most often acknowledged:

- Faster application development at a lower cost
- Decreased maintenance time
- Less complicated, faster customization of programs
- Higher-quality code

Extra benefits of OT:

- Facilitates a client/server environment
- Enables the development of a common GUI across all platforms
- Handles the storage and manipulation of more complex data and applications, such as multimedia, imaging, and groupware
- Provides the infrastructure for distributed applications

Problem: *Companies have been reluctant to build bet-the-business applications based on niche technology.*

Although the promise has been appealing, the fact is that corporate IS organizations are, by necessity, very conservative. Thus, the majority of OT choices have come from small, relatively young organizations with highly specialized solutions that address highly specialized issues. So often, when the Advanced But Very Small OT Company visits the Very Large and Conservative IS Shop, it encounters a very conscientious group of developers who would rather resort to proven approaches than risk their careers on an unknown company – regardless of the technology's potential.

Solution: *Object technology is entering the mainstream.*

Standards bodies, consortia, and large systems vendors are entering the market – all signs that OT is here to stay. Over time, this will help address other issues such as the small pool of qualified and skilled designers, architects, programmers, and assemblers; the general scarcity of standardized developer toolkits, frameworks, and class libraries; and the need for education courses, consulting, and contract engineering services.

IBM PSP recognizes that OT is strategic to the future of the computing industry. Thus, we are playing a leading role in its advancement – not only through development of our own technology, but through involvement with consortia and other key industry vendors.

Case Study: A Look Into the Future

The year is 1998. The sales manager for the western region of a large grocery store chain is opening a new store in a very competitive market. Although his new location is convenient, the manager knows his customers are very price-conscious. To make a reasonable return, he must control inventory closely. His initial plan is to exploit the infrastructure already in place:

- Point-of-sale registers that read bar codes, look up prices, and adjust store inventory records, and
- Warehouse systems that track orders and deliveries to and from major suppliers nationally as well as locally.

Although these tools provided a competitive advantage when they were installed originally, the sales manager knows that, today, even the family-owned grocery store down the street is using the same technology. So he is preparing to turn up the heat by:

- Adding "price scouts" who use mobile devices to log competitive prices and radio them into a central system, and
- Developing a customer information system that will track customer preferences to help guide future purchasing of high-margin items such as pre-prepared delicatessen foods.

It took years to design, build, test, debug, and roll out the first retail store systems – especially those designed for just-in-time delivery. Issues such as compatibility with existing systems hardware, systems software, and networks had to be negotiated by a systems integrator. Then, custom applications had to be developed and tested. Once the system was designed, adding a new store still meant extending the network, updating tables, and adding code to recognize the new store in countless applications.

With the full deployment of OT and an industry full of compatible, standards-based objects and frameworks, designing this new system won't be nearly as difficult. The radio-based personal communicator will come with software that registers the device according to OMG's CORBA standard. As a result, the device will be accessible from anywhere in the network, regardless of the physical network or software installed. Also included in the system will be a set of objects – predefined interfaces – that specify standard services, such as creating summary reports. These objects then can be modified easily to check for apparent price anomalies, so that corrections can be entered on the spot.

In a parallel process, someone from IT will browse through several visual, on-line libraries to find the mobile, competitive-pricing framework that most closely resembles their planned implementation. Another team member might search for a customer preference-tracking framework. Both frameworks will be customized with self-contained extensions to make them accurately reflect this particular grocery store, the data being gathered, and the store's customer profile. This modification will be done simply and quickly, using a new class of visual development tools.

The speed advantages of developing applications with OT are important – but far more important will be the synergistic effect of a development environment that builds on the skills and knowledge of both business and technical people. Business analysts, line managers, and developers will use common tools and languages to analyze and model the business – how IT is developed and managed, as well as how new applications can be used to make a business more competitive. Evaluating price changes, the financial projections for a new store, the effects of a merger, or deployment of a new technology – all the modelling and design can be done using visual tools that group information in a way that emulates the business itself.

PSP's Strategy for Object Technology Adoption

While many of the potential inhibitors of OT have been addressed, a lot of infrastructure still must be built to make the promise of this grocery system a reality. IDC defines the following classifications of OT products and services:

- *OO Analysis/Design Methods and CASE Tools* refer to products that support one or more methods of OO analysis and design.
- *Environments for Development of OO Applications* refer to application development environments that reduce coding. Typically, these are integrated development environments that target client/server applications and use visual programming techniques.
- *Component Software Class Libraries* refer to libraries of reusable objects and frameworks.
- *OO Programming Environments* refer to compilers, programming tools, and complete development environments for programmers writing in OO languages like C++ and Smalltalk.
- *Object Database Management Systems* refer to database methods that support object storage and retrieval.
- *Distributed Object Management* refers to the basic systems and network services that allow objects to interoperate across a distributed environment.

IBM is addressing each of the elements identified by IDC. In some areas, we have active research and development under way, resulting in specific IBM product offerings. In other areas, we are partnering with other OT leaders to help create a robust set of offerings that support industry standards. This approach is very much in keeping with IBM's OT strategy.

PSP's Strategy for Object Technology: PSP's strategy for OT is straightforward: to play a leading role in the development of an OT environment that will change the economics and simplify the process of software development for our customers, as illustrated in Figure 4.

Our efforts are guided by four underlying principles:

- We will take a leadership role in research and development.
- We will participate actively in the development and adoption of industry standards.
- We will complement the work of other leading object technology companies through key alliances and partnerships.
- We will simplify the integration of object technology into our customers' operations.

Shaping an Object Technology Infrastructure: When one looks around the industry, the collective expertise quickly becomes apparent. A vibrant collection of innovative companies provides languages, applications, and tools. Equally apparent, however, is the need for the systems structure and standards that will unify these technologies. This infrastructure must provide an environment for object-based, distributed client/server implementations – implementations that are compatible

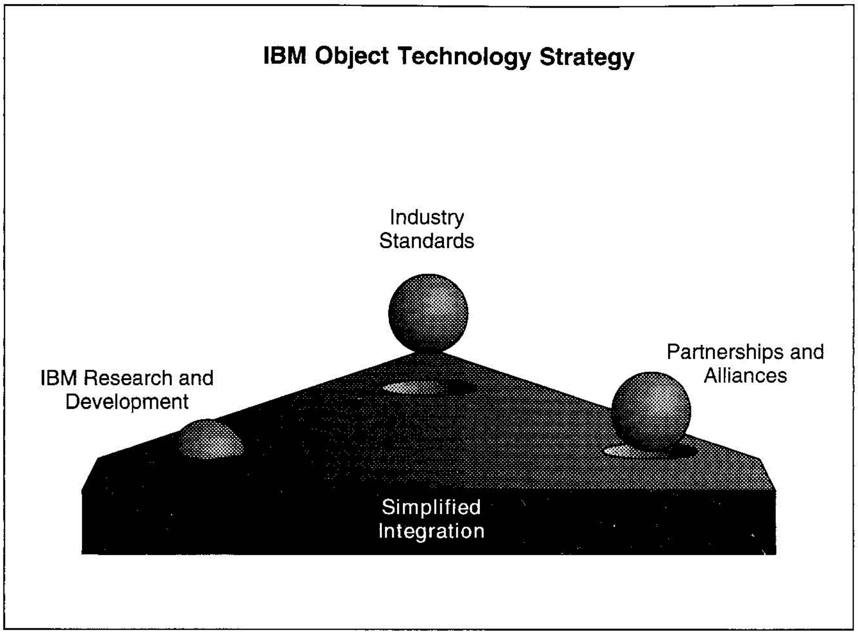

IBM Object Technology Strategy
Industry Standards
IBM Research and Development
Partnerships and Alliances
Simplified Integration

Figure 4. Four Underlying Principles of PSP's OT Strategy

OO Analysis and Design Tools

OO Application Development Environments

Component Software Class Libraries/Frameworks

OO Databases

Programming Environments

Distributed Object Management

Figure 5. Creating an OT Infrastructure

regardless of the language, platform, or network used. PSP is actively developing and delivering the enabling technology that is fundamental to building that industry infrastructure. (See Figure 5.)

PSP's Object-Enabling Strategy: The System Object Model (SOM), the cornerstone for IBM's offerings for the emerging multiplatform, distributed object computing environment, is available today in IBM's SOMobjects* Toolkit. SOM defines an infrastructure for sharing objects. It allows objects to be packaged in a way that exposes only their interfaces. As a result, an object can be written in one language and used or refined by another. For example, a component written in IBM's C++ is usable and easily refined by Digitalk's Smalltalk. SOM addresses the pervasive need for cross-language support in object development.

In addition, SOM scales up gracefully to support objects that are distributed across a network. The interface for a distributed object is identical to the interface for a local SOM object. The SOMobjects Toolkit contains Distributed SOM (DSOM), an ORB that complies with the CORBA standard set by the OMG. DSOM services locate the remote object, and route requests and responses, masking this complexity from the developer and end user. Performance is optimized as well. DSOM supports all of the important industry network transports: TCP/IP, Novell's IPX**, and NetBIOS. It will also run on the Open Software Foundation's (OSF's) DCE** network services when they become available.

IBM plans to make SOM and its distributed object extensions available on all major platforms. SOM and DSOM are packaged together in the

SOMobjects Toolkit, which is available today for OS/2* and AIX*. Future availability will include Windows**, MVS*, OS/400* other key industry platforms. The SOMobjects Toolkit will evolve to include new features, including evolving CORBA specifications.

SOMobjects is driving some very innovative work that will create a far more powerful and responsive application environment for customers. One example is Component Integration Laboratory (CILab), an industry initiative that has brought together Apple**, Oracle**, Novell**, Sun**, Xerox**, WordPerfect**, IBM, and Taligent**. Together, these companies are implementing a cross-platform architecture called *OpenDoc*. This architecture will simplify creation of compound documents – documents that include text, graphics, images, video, and sound, for example. Through OpenDoc, the elements of a compound document can come from a variety of sources, allowing a user to build a powerful multimedia document without having to create and to import the various components, as is the case today. The same technology provides a standards-based way of accessing document components between languages and across networks.

Although OpenDoc may appear similar to Microsoft's Object Linking and Embedding (OLE), industry-standard SOMobjects interfaces ensure that customers will be able to use their favorite application software on their favorite platform. OLE provides this capability only for the Microsoft** OLE-enabled applications. OpenDoc will be available during 1994 on multiple platforms: initially OS/2, Macintosh** System 7, NetWare**, and DOS/Windows. OpenDoc will interoperate with Microsoft's OLE and the next-generation, document-centered programming model from Taligent.

Distributed Computing Environment (DCE): As IBM's DSOM and the CORBA specifications evolve, they will address the major issues of network-wide security and directory systems for objects. Similar services are provided today by the Distributed Computing Environment (DCE) from the OSF. IBM and Hewlett-Packard** are sharing technologies so that their ORB will exploit DCE.

Common Operating System Environment (COSE) IBM is working with SunSoft and Hewlett-Packard to deliver a compatible ORB to the participants in the COSE definition process. Vendors of UNIX-based operating systems, including IBM, Sun Microsystems**, Hewlett-Packard, Univel, DEC**, and SCO**, are using the COSE process to define consistent programming interfaces to UNIX-based platforms. The ORB will enable consistent implementation of OT across UNIX-based system environments.

Programming Environment: With the industry's acceptance of SOM, it becomes critical that SOM be incorporated in a new generation of programming languages, compilers, and programming tools. A critical mass of OO vendors has embraced SOM, and is working to shape a more powerful, flexible OO programming environment:

- IBM and MetaWare will deliver Direct To SOM (DTSOM) C++ compilers. This means these compilers will generate SOM objects automatically.
- Digitalk has announced SOM support in its Smalltalk/V** development environment.
- ParcPlace has stated its intention to support SOM in its Smalltalk development environment.
- WATCOM has stated its intention to support SOM in its C++ development environment.
- MicroFocus** has stated its intention to support SOM in its OO COBOL environment.

Discussions with other vendors are under way as well.

Object-Oriented Database Management Systems: OO databases are critical to the success of advanced distributed computing environments. A number of companies, including Object Design and Ontos, have focused on this growing segment, and have committed to SOM support. IBM's strategy for object-oriented database management systems (OODBMS) is twofold:

- In partnership with Object Design, we will provide OODBMS services for IBM tools, and a general OODBMS.
- We will also provide general-purpose enabling services for other OODBMS.

Component Software Class Libraries and Frameworks: IBM's SOMobjects Toolkit provides an infrastructure for distributed software components or objects. Frameworks represent groups of compatible software components. In other words, frameworks capture the collected experience of a design team. They address a particular problem, and are common in most advanced OT products. Their value is that they

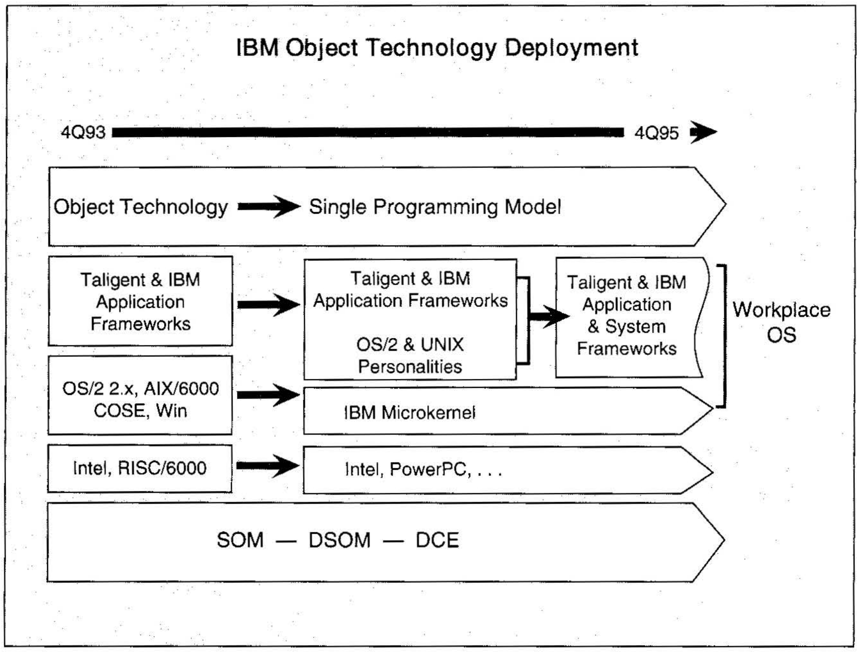

IBM Object Technology Deployment
4Q93
4Q95
Object Technology
Single Programming Model
Taligent & IBM Application Frameworks
OS/2 2.x, AIX/6000 COSE, Win
Intel, RISC/6000
Taligent & IBM Application Frameworks
OS/2 & UNIX Personalities
IBM Microkernel
Intel, PowerPC, . . .
Taligent & IBM Application & System Frameworks
Workplace OS
SOM — DSOM — DCE

Figure 6. OO Technology Deployed Over Existing and Evolving Industry-Leading Hardware Platforms

automate most of the development effort in a particular area, such as designing and implementing a GUI. IBM will deliver an increasingly rich set of frameworks across key industry operating systems and operating environments, including OS/2, AIX, Windows, the COSE environment, and Workplace OS*, in five stages:

1. Frameworks that automate some of the tasks associated with distributed applications

2. Frameworks that mask the differences between operating systems, allowing applications to be ported easily among them

3. Taligent desktop frameworks that will radically alter the economics of building advanced graphical applications

4. Taligent and IBM network frameworks that deal with complex transactions and communications

5. Taligent and IBM application and system frameworks as Workplace OS – Taligent personality

These frameworks will simplify deployment of distributed, component-based, collaborative applications, as shown in Figure 6.

The IBM/Taligent frameworks will provide advanced object application programming interfaces (APIs) for both new and evolving applications. These frameworks will provide the same APIs and advanced functions across all platforms. Developers will be able to adopt the new framework-based services at their own pace. For example, new applications may be written entirely to the object services, while existing applications can take advantage of the new services as they evolve to meet new requirements.

IBM PSP shipped the SOMobjects Toolkit with several frameworks that help developers build distributed applications. Next, PSP will provide a set of portability frameworks and classes that can substantially isolate a developer from platform differences, everything from memory management to advanced graphics. This is the first set of frameworks that really begins to address the problems of multiplatform object development.

The result of 1994's product deliveries will be a set of complementary frameworks that provide a new programming model for software, an IBM Taligent-based portable application environment. Rather than creating large applications, developers will be able to make small extensions to the frameworks. The environment ensures that these extensions automatically and seamlessly integrate with the framework, and each other, to exchange data and to use each other's services. This portable application environment is an OO development layer that will be made available across all key industry OS platforms, including OS/2, Workplace OS, AIX, DOS/Windows, and others.

Key frameworks in this category will be the document, document user interface, and desktop frameworks, as well as a set of generic tools for test, graphics, image editing, and database access.

These broad categories of frameworks will allow developers to incrementally evolve their teams and software to take advantage of OT while preserving their investment in existing procedural application functions and skills.

When the IBM Taligent-based portable application environment programming model is used, a new style of collaborative, distributed, independently extensible software can be created with a fraction of the programming required today. This is also the application programming model of the future Taligent OO operating system, which extends the use of frameworks into operating system services. IBM will ship this as the Taligent personality for Workplace OS. All the interfaces defined as part of the portable application environment will be supported in the OO operating system frameworks. Therefore, applications based on the portable application environment can be recompiled to run on any vendor's implementation of the Taligent operating system or other operating system supporting the portable application environment, again preserving the investment's application function. Rollout of these Taligent frameworks began at the end of 1993.

Object-Oriented Application Development Environments: These environments represent a significant growth area for OO deployment. In this area of OO development environments, IBM is actively working with key independent software vendor (ISV) partners to gain their support for IBM's distributed object computing environment (SOM/DSOM). These partners include vendors such as Digitalk, EASEL** Corporation, and Inference Corporation. In addition, IBM has announced VisualAGE*, a visual programming environment based on the Smalltalk programming language. This powerful tool, geared primarily for the professional programmer, simplifies the development of client/server applications.

Object-Oriented Analysis and Design Tools: OO analysis and design represents a new and emerging discipline for OO development. It is based on the required transitional shift from existing, proprietary CASE implementations to those based on the object paradigm. IBM's strategy is to continue enabling CASE tools vendors to ensure that their tools are able to interact with existing IBM relational database management systems and, over time, to support object-oriented database management systems.

Object technology provides a powerful new vision of programming.

Getting Started with Object Technology

Object technology provides a powerful new vision of programming, and it is not too early to begin exploiting it. Using OT requires some training and the adoption of new tools, but implementation can be gradual – starting with a small team of individuals and a simple project. The choice of tools will vary, depending on the applications being developed. There are two distinct categories identified by IDC: OO application development environments, and integrated language environments. The former is more suitable to groups developing business applications with skills in high-level languages like COBOL. The latter will be more appropriate for groups undertaking system development with lower-level languages like C or C++.

OO application development environments, such as (but not limited to) IBM's VisualAge, Digitalk's Smalltalk and PARTS products, Intelligent Environments' Application Manager, ParcPlace, and Systems VisualWorks, provide sophisticated tools for building client/server applications that span networks and systems. They focus on powerful tools for developing user interfaces and business logic on PCs and workstations. These tools also provide functions to link to existing transactional and database systems like IBM's Customer Information Control System (CICS*) and DATABASE 2/2 (DB2/2*). All of these products, any many others, offer several distinct advantages to companies planning for OT. They encourage and reward an objects-based approach to design; they offer an integrated, high-level environment that can speed training; and they work well with existing applications and systems.

Developers using lower-level languages or developing advanced subsystems likely will be better served by the integrated C++ development environments. C++ offers an OO extension to the C language, and both IBM and MetaWare offer versions that provide native support of IBM's SOM. These products combine the development functions for compiling, program editing, and debugging in one toolset, but they rely on the developer to provide higher-level constructs or access to other systems and databases.

The Smalltalk products from ParcPlace, Digitalk, and EASEL bridge these two separate worlds, and may be the right choice for programmers who require flexibility, but are unwilling to take on the challenges of lower-level languages like C++. While these environments may not offer the productivity of the application environments, they are more flexible and offer more control over application size and performance.

American Management Systems

Borland International, Inc.

ChipChat-Cawthon Software

Cirrus Technology, Inc.

Continuum Company, Inc.

Digitalk, Inc.

EASEL Corporation

Footprint Software, Inc.

Hewlett-Packard

Inference Corporation

Information Advantage, Inc.

Intermedia Development Company, Inc.

KASEWORKS, Inc.

MetaWare, Inc.

Microformatic

Object Design, Inc.

Raleigh Systems, Inc.

Sundial Corporation

SunSoft, Inc.

WATCOM International Corporation

Figure 7. ISVs Who Plan to Support SOM

IBM's SOMobjects Toolkit offers some advanced development functions to users of these environments. The lower-level tools will support SOMobjects and offer programmers access to the detailed function it provides. The application environments generally hide the details of SOMobjects, simply surfacing its value by allowing objects to be accessed across a network in a visual development tool.

As the object frameworks are delivered, they will be exploited in this fashion. The underlying details in the language environments are hidden but exploited in the higher-level application tools.

A business's chances of successfully deploying OT can be increased with OT education. Taligent's advice to developers interested in OT adoption is:

- *Learn OO design.* Developers who only use C++ as a better C, without fundamentally changing their design approach to use encapsulation, polymorphism, and inheritance (all functions of OT), will not realize the substantial benefits of OT.
- *Learn an OO language, and begin to use it exclusively.* IBM, Intelligent Environments, MetaWare, Digitalk, EASEL, ParcPlace, WATCOM**, Borland**, and Micro Focus are examples of vendors who offer OO languages and environments, and who support or plan to support SOM/DSOM in the near future. Figure 7 lists many of these vendors.
- *Learn to design and work with frameworks.* Begin with class libraries, and understand both their power and their limitations. Then begin to think in terms of frameworks, becoming familiar with commercially available frameworks from vendors such as Footprint or ChipChat**.

Finally, while strong management commitment to OT and careful selection of tools are important, consultants and education specialists can be the difference between success and failure. While many specialize in OT, a complete list of educators is well beyond the scope of this article. However, many OT providers also offer consulting and education, including IBM Consulting Practices, Digitalk, ParcPlace, Knowledge Systems, Raleigh Systems, and Anderson Consulting.

In addition, Skill Dynamics, an IBM company, offers courses in OO concepts, management, analysis design, languages, database, and user interface. For a detailed list of course descriptions, pricing, and scheduling information, contact:

Object-Oriented Curriculum
Manager
Skill Dynamics
6 East 55th Street
New York NY 10022
1-212-230-5056

To be added to the Skill Dynamics OT mailing list, call 1-212-230-5440.

D'Ann Ostrom is Taligent Brand Manager within the IBM Personal Software Products marketing organization in Austin, Texas. She is responsible for developing marketing plans for IBM PSP object-oriented products.

OpenDoc: An Idea Whose Time Has Come!

Robert Tycast
IBM Corporation
Boca Raton, Florida

This article is reprinted from Volume 2 of The Developer Connection News, *sold by subscription, together with The Developer Connection for OS/2 CD-ROM, $199 (U.S.) per year for four quarterly releases.*

When Brad Cox coined the term "software IC," he foresaw a day when software building would move from the realm of hand-crafted monoliths to a world of composites. These new-age solutions would be assembled from parts by skilled application builders, using and reusing software components in ways that the original designers didn't and couldn't have conceived of.

OpenDoc represents an important first step in that direction. It provides the technology necessary to break up an application into parts.

OpenDoc – What's in a Name?

"*Open*" – OpenDoc is a programming architecture for creating, storing, and sharing compound documents. It is open, vendor-neutral, language-independent, and cross-platform.

Born of work by Apple Computer, Inc., and supported by major vendors such as IBM, Novell, Oracle, WordPerfect, Xerox, and Taligent, OpenDoc is changing the way applications are built and used.

"*Doc*" – OpenDoc is document-centered programming. By a suitable change of mind-set, virtually all applications in use today can be seen as a *document*. And, the definition is expanded. In OpenDoc, documents include more than text – audio, video, graphics, charts, spreadsheets – virtually anything that a computer can output is fair game. The document must be alive and not static; animation, background music, and a dynamically changing content are all part of the OpenDoc document.

Comparing OpenDoc with OLE2

OpenDoc	**OLE2**
Open system, freely licensed to the industry through the Component Integration Laboratory.	Proprietary, owned and controlled by Microsoft.
SOM, based on industry standard for object-oriented programming (CORBA).	COM, *not* CORBA-compliant; no inheritance; aggregation proposed as alternative.
Distributed – OpenDoc parts can be embedded from anywhere in the network.	Not distributed – can only imbed objects from local OLE servers.
Cross-platform support – OpenDoc will be available on Apple, OS/2, UNIX, and Microsoft Windows.	Only available on Microsoft Windows.
Language-Neutral – SOM bindings make OpenDoc readily available from any language.	Difficult to use with languages other than C++.
Source code available.	Source code *not* available.
Any part can be at the "root" of the document.	Only specialized containers can be at the "root."
Parts can be any shape.	OLE objects must be rectangular.
OpenDoc maintains multiple draft versions of a document.	No support for multiple drafts in OLE.
OpenDoc parts can overlap.	OLE objects *cannot* overlap.
OpenDoc parts can be edited by clicking on them directly.	OLE objects must be activated and the content selected in order to edit it; when nested, multiple levels have to be activated.

You will no longer decide which applications to launch to solve a problem or to do work on your computer. Instead, you'll start with blank stationery, and compose a document by collecting and combining "parts in standard or novel ways, depending on your need, inclination, or experience level."

The most simple documents, ones that traditionally would use a text editor or word processor, simply will include a text part. If, however, you wish to spice up your memo or letter with some graphics, simply include a graphics part using your favorite graphics editor. Or, if you want to include up-to-the-minute sale data, then a bar graph linked to a spreadsheet will do the trick.

Contrast this with an application-centered model, where you can't decide to include graphics in your document as an afterthought. You have to choose a word processor which has *all* of the capabilities that you will eventually want.

With OpenDoc, you are limited only by your imagination, and not by the capabilities of the application selected.

Is That All?

Not by a long shot! OpenDoc documents are scriptable. That means that developers can provide unique solutions by gluing OpenDoc parts together with a script language such as ObjectREXX. And users benefit, because they can tinker and customize their documents to their heart's content.

An Amalgam of Technologies

OpenDoc will consist of four distinct technologies:

- *Compound Documents.* This is OpenDoc proper.
- *System Object Model (SOM).* SOM provides the CORBA-compliant capabilities of OpenDoc, including a language-neutral interface and the ability to access parts across a network.
- *Open Scripting Architecture.* OSA provides the ability to script a document at the part level.
- *Bento.* This is the persistent storage model. It is available to developers to use to write OpenDoc documents to permanent store.

Making These Technologies Available to the Industry

Enter CILab. Apple, IBM, Novell, Oracle, WordPerfect, Xerox, and Taligent have agreed to license these technologies to a jointly-funded consortium. To that end, Component Integration Laboratory (CILab) was founded. Component Integration Laboratory is a non-profit association dedicated to software plug-and-play interoperability across multiple computer platforms.

Modeled after the successful X Window System** Consortium, CILab will receive the rights to OpenDoc, SOM, and related technologies. It will then license them back, royalty-free, to the industry. Members will contribute reference implementations and other donated software. CILab will also develop certification programs, as a service to the industry, to verify the completeness and correctness of OpenDoc implementations, as well as offer training for developers who want to use CILab technologies.

***Robert Tycast** is an advisory programmer in the OS/2 Development group. Over the last 15 years, Robert has had project experience in X11, AI Technology (LISP and OPS5 support), and technical workstations (VMS and ULTRIX).*

CILab Brief

This is a brief introduction to Component Integration Laboratory (CILab), including its background, purpose, and goals.

Transition to Software Components

Software developers want to create applications more quickly and deliver more functionality. Users want more control over the applications they use and the documents they create. Everyone wants to support multiple platforms and access to distributed information and services.

Because of these needs, the industry is ready for a move to software components. Using software components, users can build compound documents that seamlessly integrate text, graphics, tables, multimedia, scripts, and other forms of content. In effect, sophisticated users can build custom applications.

At the same time, software components allow software developers to focus on their competitive advantage, while providing a richer feature set by bundling multiple components. This allows them to leverage OEM software opportunities, and also to develop new business opportunities based on vertical bundles and a wider range of upgrade paths.

The technology to make this move to software components is here today. Unfortunately we are starting to see the signs of a familiar problem – multiple incompatible technologies, potential market fragmentation and awkward choices for developers and users.

Providing a Reliable Foundation

A group of companies – Apple, IBM, Novell, Oracle, Sun, Taligent, WordPerfect, and Xerox – have come together to organize the Component Integration Laboratory (CILab) as an industry association that will provide a common foundation for software components.

CILab is not a standards organization. Instead, the founders plan to have it adopt, maintain, license, and support essential software component technologies, such as object dynamic linking, object storage, scripting mechanisms, and compound document APIs.

By providing reference source code for these foundation technologies, the Lab can make sure that a common software component architecture is rapidly implemented across all the major industry platforms, including Microsoft Windows, Macintosh, OS/2, and various UNIX** systems.

Foundation Technologies

The founders are planning to start CILab out with a very complete set of foundation technologies:

- The System Object Model (SOM), a highly efficient object dynamic linking mechanism, which supports multiple languages and provides a gateway to distributed object services.
- Bento, a portable object storage library and format designed for the storage and interchange of compound documents and multimedia.
- The Open Scripting Architecture (OSA) an automation and scripting API that supports application independent scripting, distributed automation, and workflow applications.
- OpenDoc, a platform independent compound document architecture that supports integration of multiple software components into seamless documents and custom applications.

Three of these initial technologies are already available from their developers: the System Object Model from IBM, and Bento object storage and the Open Scripting Architecture from Apple. IBM and Apple have announced their intent to provide these technologies to CILab in early 1994.

The compound document API, OpenDoc, is being implemented in parallel by Apple, IBM, WordPerfect, and other companies, and these companies plan to transfer it to CILab when it is complete, in late summer of 1994.

In addition to these initial technologies, over time CILab plans to adopt other technologies that enrich the industry-wide component software foundation. Several companies have already initiated discussions with the Lab regarding the possibility of donating specific technologies.

Membership in CILab

We are planning to open CILab for general membership in early 1994. Lab members will gain participation in decisions and early access to technology. In addition, over time we are planning to provide a wide range of services to members, including certification to ensure interoperability, developer support, training, and co-marketing.

We are interested in talking to potential members to determine what technologies and services would have the greatest value to them as members of CILab.

Contacting CILab

We'd be happy to answer further questions about CILab. If you would like to talk with us further regarding our plans, please contact us.

Email: cil@cil.org
Voice: 1-415-750-8352
Fax: 1-415-751-4829
US Mail: Component Integration Laboratory
688 Fourth Avenue
San Francisco CA 94118

Technical Update, Part 3: OS/2 2.1 Hardware Support

This article is excerpted and adapted from the OS/2 2.1 Technical Update, *one of a series of Red Books published by the IBM International Technical Support Center (ITSC) in Boca Raton, Florida. The IBM order number for this publication is GC24-3948. The first two parts of this article appeared in Issues 1 and 2, 1993 of* IBM Personal Software Technical Newsletter. *Part 3 covers hardware support in OS/2* 2.1, including BIOS, diskette drives, rewritable optical drives, device drivers, disk and SCSI adapters, SCSI-based CD-ROMs, video, laptop computers, and notebook computers.*

OS/2 2.0 provided an advanced 32-bit operating system for PCs, based on the Intel 386** and 486** microprocessors. It was architected to support and exploit a wide range of hardware, through the use of installable device drivers.

OS/2 2.1 extends that hardware support by including additional SCSI device drivers, SCSI-based CD-ROM device drivers, and video device drivers in the OS/2 2.1 package. Additional device drivers will be available from hardware device manufacturers and independent software developers, and also on bulletin boards.

In addition, OS/2 2.1 also includes changes to exploit Intel's Pentium** chip, through the use of the Pentium virtual-mode extensions.

Industry Hardware Support for OS/2 2.1

OS/2 2.1 has been tested on computers from a wide range of PC manufacturers (PCMs). This testing was done by the PCMs, by IBM at its test laboratories in Boca Raton, Florida and Basingstoke, UK, and through two widespread beta tests of OS/2 2.1.

OS/2 2.1 has also been tested in conjunction with many PC adapters and peripherals from a wide range of OEM manufacturers and independent hardware vendors (IHVs). This testing was done by the OEMs and IHVs, by IBM, and in the beta tests.

The current list of IBM, PCM, OEM, and IHV hardware supported by OS/2 2.1 is in the PCMTABLE package available on CompuServe** and within IBM in OS2TOOLS.

Understanding OS/2 2.x Hardware Support

OS/2 2.x supports a wide variety of hardware by insulating the operating system from the hardware. This is achieved through the use of installable device drivers and the BIOS firmware. New hardware devices can be supported by providing a new device driver, or by providing a standard BIOS interface on top of the new hardware.

This hardware abstraction layer has evolved many times since OS/2 1.0, and today it contains a variety of device driver and BIOS approaches. A simplified view of the hardware support components of OS/2 2.1 is shown in Figure 1.

Hardware support in OS/2 2.1 is provided by a combination of the BIOS and device drivers. Most of this support also applies to OS/2 2.0.

BIOS

The BIOS provides the lowest level of the OS/2 2.1 operating system, and is normally contained in ROM on the system board. Although the BIOS is software, it is provided by the hardware manufacturer along with the PC.

The ROM BIOS is assigned the address range F000-FFFF (E000-FFFF on PS/2* computers). Originally introduced with the DOS operating system and the first IBM PC, the BIOS provides a standard set of basic functions for input/output devices such as the keyboard and the disks.

All IBM PCs and compatible computers contain a BIOS. Although many DOS applications avoid using the BIOS for some functions (especially video), it is normally safe to assume that the BIOS is present. In IBM PS/2 computers, the original BIOS used by DOS is referred to as CBIOS (for Compatibility BIOS).

IBM PS/2 Micro Channel* computers also contain an ABIOS (for Advanced BIOS). It was designed to provide a BIOS optimized for use by multitasking operating systems such as OS/2. ABIOS is also normally held in ROM, and is mapped into the operating system address space (between 640 KB and 1 MB), but in some recent PS/2 models the ABIOS is held instead on disk, and is loaded into RAM.

OS/2 2.1 uses the ABIOS if present; otherwise, it uses the BIOS. For some input/output functions, OS/2 2.1 also accesses the hardware directly (through the device drivers).

Device Drivers

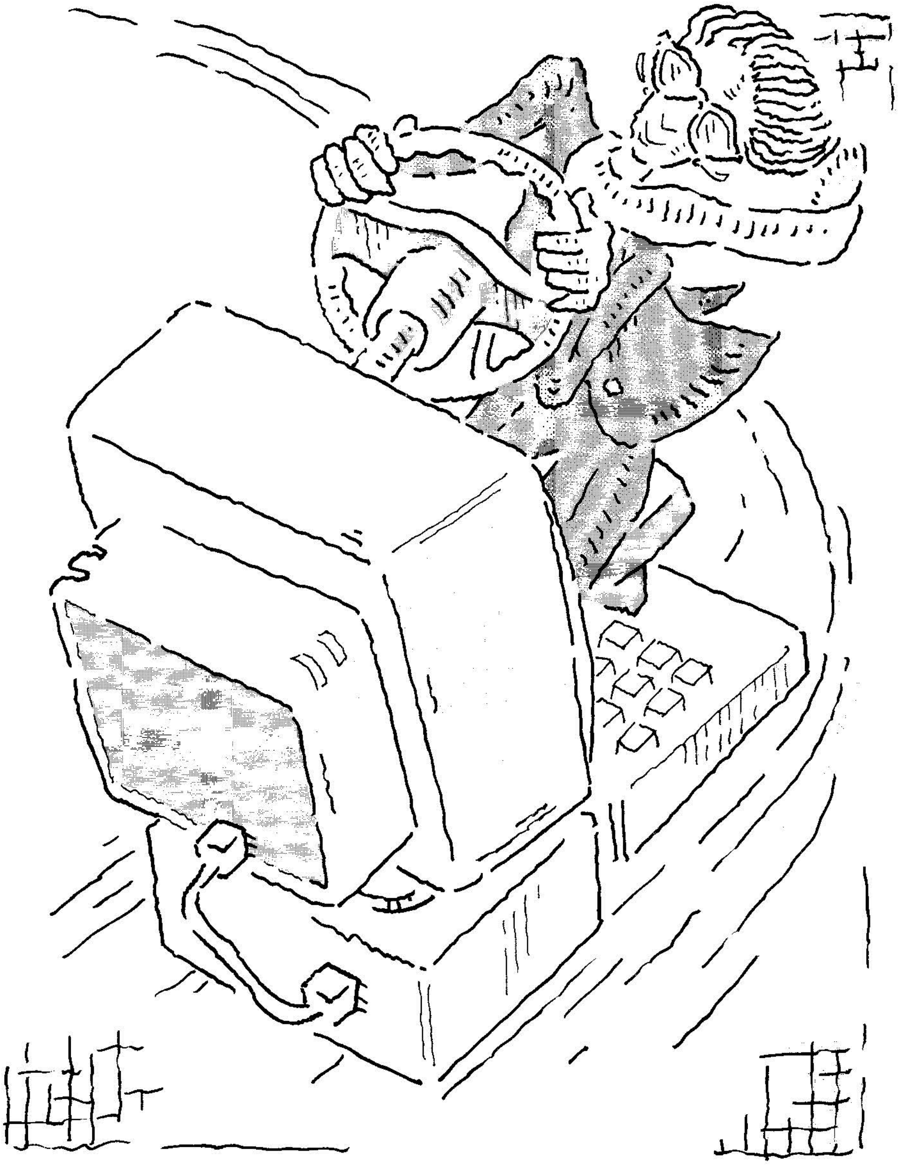
The other layer of software that insulates the kernel of OS/2 2.1 from the hardware is the device-driver layer.

Device drivers were originally provided as *.SYS files, typically one for each hardware component, which were loaded during the operating system's initialization.

There are now a variety of device-driver types and flavors in OS/2 2.1. Disk and SCSI device drivers have now adopted a layered device-driver structure, which reduces the effort needed to support a new hardware device. Video and printer drivers also have a specialized module structure.

Most device drivers are specified in the CONFIG.SYS file, using the BASEDEV= or DEVICE= keywords. The BASEDEV device drivers are needed for OS/2 2.1 to run, and are loaded first, followed by the DEVICE device drivers. These drivers are sometimes called Physical Device Drivers (PDDs).

Corresponding to the PDDs are Virtual Device Drivers (VDDs), which provide device support for DOS and Windows applications running in the MVDM and WIN-OS/2* environments. VDDs interface to the physical device drivers, rather than accessing the hardware directly.



Loadable ABIOS Support

The original ABIOS implementations in PS/2 Micro Channel systems were in ROM; thus, ABIOS could be assumed to be always present (in the same way that DOS assumes that the BIOS is always present).

When changes had to be made to the BIOS, they were implemented as ABIOS patch (*.BIO) files. These patch files were listed in the ABIOS.SYS file. ABIOS patch files included both general patch files and patch files for specific PS/2 models.

The ROM ABIOS implementation on IBM PS/2 systems stored the ABIOS code in the same 128 KB of ROM where the POST (Power-On Self-Test diagnostics) and the BIOS (CBIOS) code are located.

As PS/2 systems became more advanced, some systems literally ran out of room in the 128 KB of ROM to store all of POST, BIOS, and ABIOS. It has thus become necessary to move the ABIOS out of the ROM and into a file on the disk, which can be loaded into RAM as part of the operating system. The ABIOS on disk is known as *loadable ABIOS.*

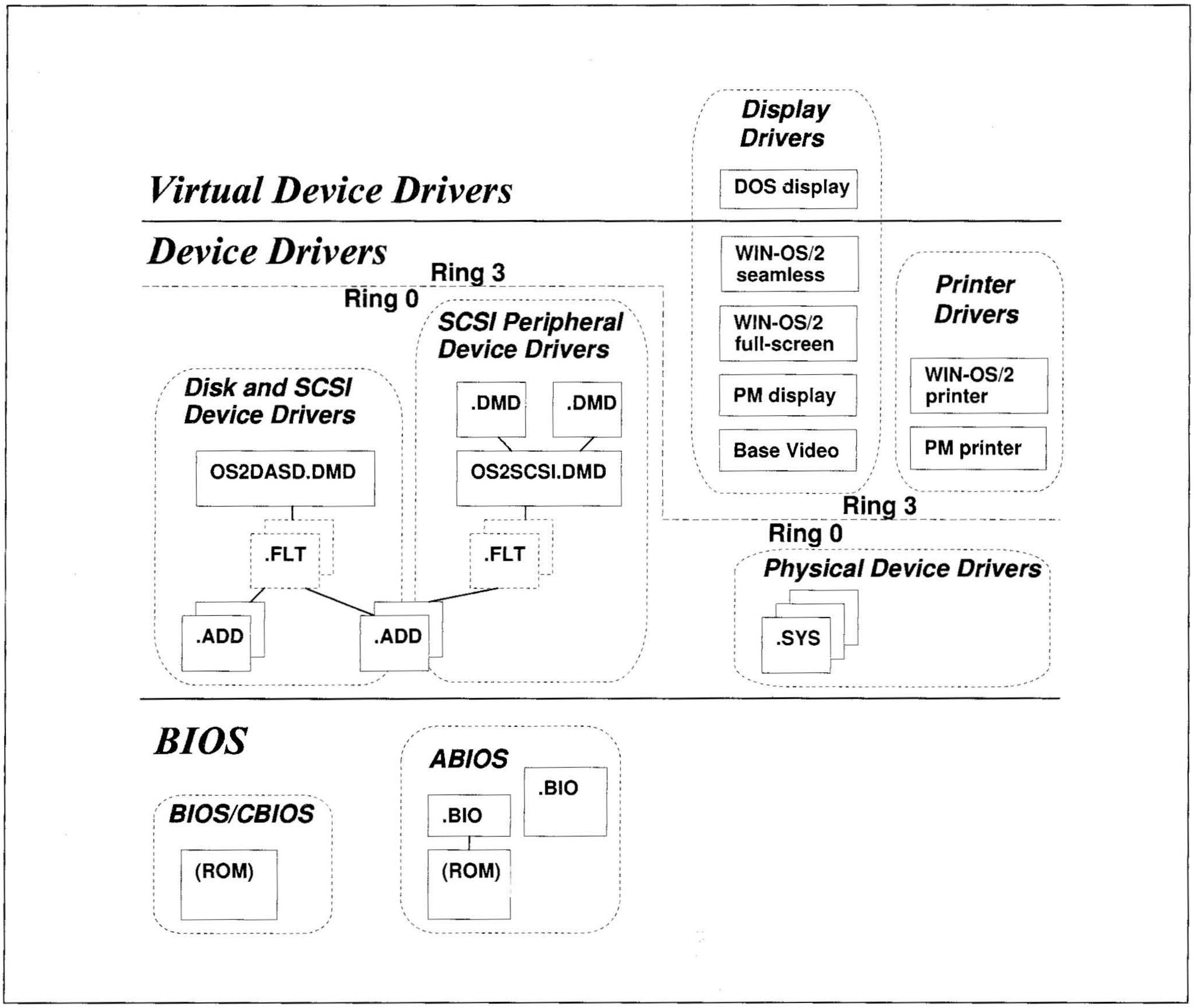


Figure 1. Overview of OS/2 2.1 Hardware Support Layer

Loadable ABIOS has been implemented on the following systems:

- PS/2 models 56 and 57
- PS/2 Server 85
- ThinkPads* 700 and later

Loadable ABIOS is supported by OS/2 2.1, OS/2 2.00.1, and OS/2 2.0 with ServicePak XR6055 applied.

The introduction of loadable ABIOS has implications for OS/2 2.x. Both OS/2 2.0 and 2.1 depend on the presence of ABIOS to run on IBM PS/2 Micro Channel systems (although they both run on IBM and PCM computers using a mixture of BIOS and direct hardware support). Hence, OS/2 2.x needs to:

- Know which ABIOS files to load
- Install the appropriate ABIOS files during OS/2 2.x installation

The ABIOS files are specified in the ABIOS.SYS file, the same way that ABIOS patch files are specified. However, in this case it is the complete ABIOS that is being loaded.

The OS/2 2.1 installation procedure has been enhanced to install the appropriate ABIOS file on the fixed disk. The ABIOS file is copied from the system partition using a special device driver; if this is not possible, the user is prompted for the Hardware Support Diskette. At this stage, the PS/2 Reference Diskette should be inserted. The

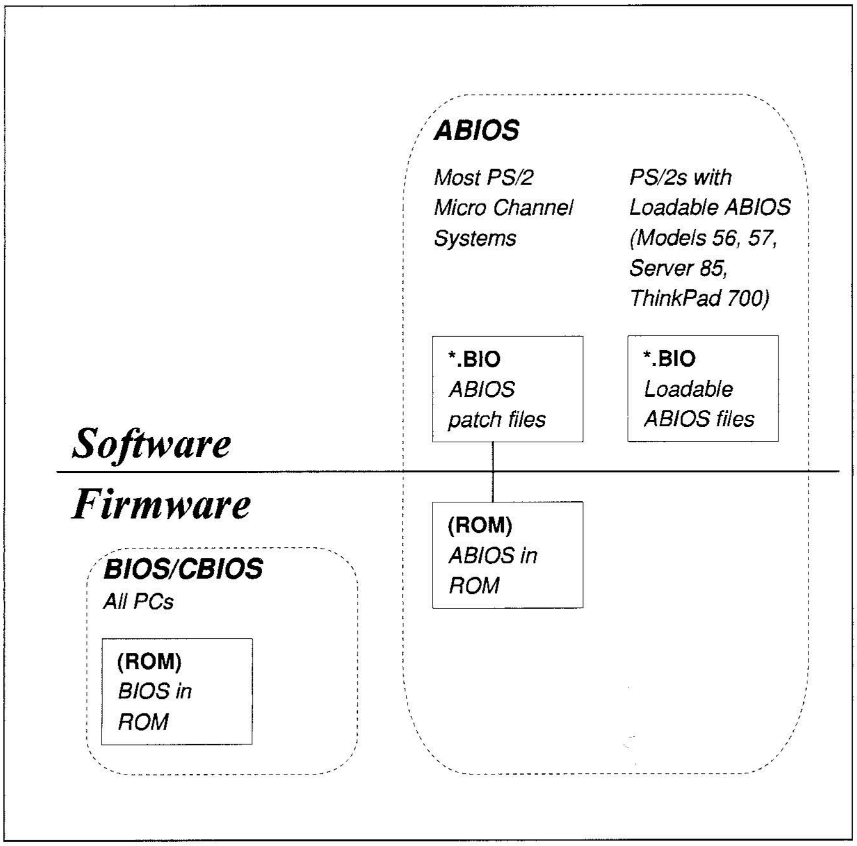

ABIOS
Most PS/2 Micro Channel Systems
PS/2s with Loadable ABIOS (Models 56, 57, Server 85, ThinkPad 700)
*.BIO ABIOS patch files
*.BIO Loadable ABIOS files
Software
Firmware
(ROM) ABIOS in ROM
BIOS/CBIOS
All PCs
(ROM) BIOS in ROM

Figure 2. OS/2 2.1 BIOS and ABIOS Support

ABIOS.SYS file is then modified to include the ABIOS filename as its first entry.

Figure 2 depicts the BIOS and ABIOS support in OS/2 2.1.

Basic Hardware Support

The following additions have been made to the basic hardware support included with OS/2 2.1:

- Support for the PS/2 Server 195 and 295
- Support for the new Brazilian keyboard
- Support for the enhanced features of the new 2.88 MB diskette drive
- Support for the new 3.5-inch enhanced rewritable optical disk drive.

Support for the PS/2 Server 195 and 295, and for the Brazilian keyboard, are not discussed in this article. Refer to the *OS/2 2.1 Technical Update* for this material.

In addition, the following enhancements have been made to support laptop computers:

- Advanced Power Management (APM) support
- PCMCIA** support
- VGA large-cursor support on LCD screens
- Support for the Trackpoint II* pointing device

Enhanced 2.88 MB Diskette Drive Support: There are three varieties of PS/2 2.88 MB diskette drives:

- 2.88 MB diskette drive (6451106)
- Enhanced 2.88 MB diskette drive with software eject (6451272)
- Enhanced 2.88 MB diskette drive with software eject and software lock/unlock (6451271)

OS/2 2.1 includes support for all three drives, including the enhanced features for software eject and software lock/unlock where appropriate. Software eject enables the diskette to be ejected under software control, typically from the pop-up menu of the diskette icon. Software lock/unlock enables the diskette drive to be locked or unlocked under software control, again typically from the pop-up menu of the diskette icon. This provides additional security protection for the workstation and its data.

3.5-Inch Enhanced Rewritable Optical Drive Support: IBM's 3.5-inch Enhanced Rewritable Optical Drive (6451295) supports Partial Read-Only Memory (P-ROM) optical disks, as well as the Magneto Optical (MO) and Optical Read-Only Memory (O-ROM) optical disks supported by the previous 3.5-inch Rewritable Optical Drive.

All three optical disks are provided in 3.5-inch cartridges. The differences are:

- MO disks can be read and written many times in the optical drives. MO disks are usually formatted at 127 MB.
- O-ROM disks can only be read in optical drives. They must be created using special equipment (as with CD-ROMs). O-ROM disks are usually formatted at 122 MB.
- P-ROM disks are a hybrid of MO and O-ROM disks. They contain

<table>
<tr><th colspan="3">Basic Device Drivers in OS/2 2.1 that are Loaded Automatically</th></tr>
<tr><th>Device</th><th>ISA Device Driver</th><th>Micro Channel Device Driver</th></tr>
<tr><td>Clock</td><td>CLOCK01.SYS</td><td>CLOCK02.SYS</td></tr>
<tr><td>Keyboard</td><td>KBD01.SYS</td><td>KDB02.SYS</td></tr>
<tr><td>Screen</td><td>SCREEN01.SYS</td><td>SCREEN02.SYS</td></tr>
<tr><th colspan="3">Basic Device Drivers in OS/2 2.1 that are Loaded with BASEDEV=</th></tr>
<tr><th>Device</th><th>ISA Device Driver</th><th>Micro Channel Device Driver</th></tr>
<tr><td>Printer</td><td>PRINT01.SYS</td><td>PRINT02.SYS</td></tr>
<tr><th colspan="3">Basic Device Drivers in OS/2 2.1 that are Loaded with DEVICE= (same drivers for ISA and Micro Channel systems)</th></tr>
<tr><th colspan="2">Device</th><th>Device Driver</th></tr>
<tr><td colspan="2">Hardware Configuration Testing (used by hardware presence-checking programs)</td><td>TESTCFG.SYS</td></tr>
<tr><td colspan="2">Presentation Manager Draw Support</td><td>PMDD.SYS</td></tr>
<tr><td colspan="2">Mouse Pointer Draw Support</td><td>POINTDD.SYS</td></tr>
<tr><td colspan="2">Touch Devices</td><td>TOUCH.SYS</td></tr>
<tr><td colspan="2">System Error Logging</td><td>LOG.SYS</td></tr>
<tr><td colspan="2">Serial Device Support</td><td>COM.SYS</td></tr>
<tr><td colspan="2">PCMCIA Bus Support</td><td>PCMCIA.SYS</td></tr>
<tr><td colspan="2">Advanced Power Management Support</td><td>APM.SYS</td></tr>
<tr><td colspan="2">External Diskette Support</td><td>EXTDSKDD.SYS</td></tr>
</table>

Figure 3. OS/2 Basic Device Drivers Loading Sequence

a designated read-only area and a rewritable area. The whole P-ROM disk can be read and the rewritable portion written using the Enhanced Rewritable Optical Drive. P-ROM disks are usually formatted at 122 MB.

The OS/2 format utility has been enhanced to be able to format P-ROM disks in the Enhanced Rewritable Optical Drive. This drive also includes software eject and software lock/unlock features, which can be used from OS/2 2.1, typically from the pop-up menu of the optical drive icon.

Device-Driver Support

The basic hardware device drivers are provided as *.SYS files. Apart from the keyboard, screen, and clock device drivers (which are loaded automatically), these device drivers are specified in the CONFIG.SYS file in BASEDEV= statements and DEVICE= statements.

BASEDEV device drivers are loaded first, because they are needed to load the rest of the operating system. Since they are loaded early, no path can be specified, and the device-driver files must be in either the root directory or the \OS2 directory of the boot drive.

DEVICE device drivers are then loaded, and a pathname can be specified.

For a more comprehensive discussion of device-driver architectures, along with sample code, refer to the IBM OS/2 Device Driver Development Kit, which is available on CD-ROM.

Physical Device Driver Support: Figure 3 lists all the basic device drivers in OS/2 2.1, and specifies whether they are loaded automatically, or with the BASEDEV= statement, or with the DEVICE= statement.

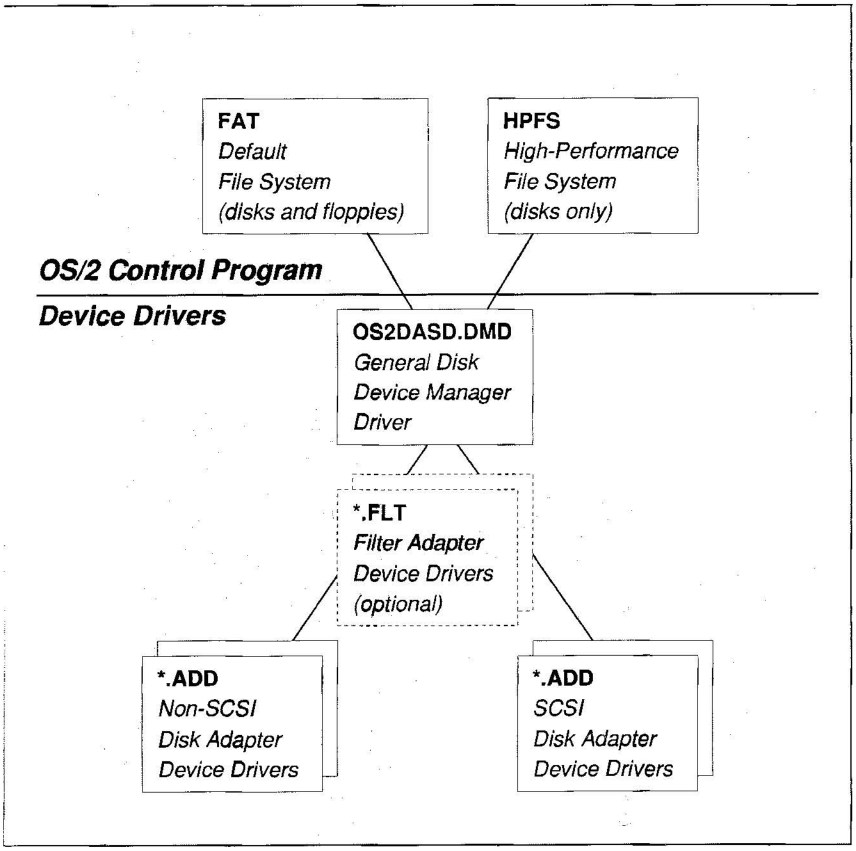


Figure 4. OS/2 2.1 Disk and SCSI Adapter Support

Disk and SCSI Adapter Support

OS/2 2.1 provides support for a number of SCSI and non-SCSI disk adapters for both ISA and Micro Channel computers. This support is provided through a combination of the hardware-specific SCSI adapter device drivers (*.ADD) and the hardware-independent disk device manager (OS2DASD.DMD). In addition, filter device drivers can be installed between the .ADDs and the .DMD to provide special functions such as encryption.

At installation time, in most cases, OS/2 senses the installed adapters, and then loads the necessary .ADD files and includes the necessary BASEDEV= statements in CONFIG.SYS. If a suitable device driver cannot be found for all the disks installed in the system, then the IBMINT13.I13 device driver is installed. This driver uses the BIOS to provide the disk support.

Figure 4 illustrates the disk and SCSI adapter support in OS/2 2.1.

OS/2 2.1 Enhancements: OS/2 2.0 included SCSI adapter support for the IBM PS/2 Micro Channel SCSI adapters, and also default support for ISA SCSI adapters, using the INT13 BIOS interface. OS/2 2.1 adds the leading SCSI adapters, including Adaptec, DPT, and Future Domain, in addition to IBM.

To find out which SCSI adapter device drivers are supplied with OS/2 2.1, look at the SCSI.TBL file in the \OS2\INSTALL directory. This file is used by the Selective Install program; it contains a list of the SCSI adapters along with their corresponding device drivers, and also the name of the appropriate hardware presence-check program. Figure 5 lists the drivers that appear in SCSI.TBL.

SCSI-Based CD-ROM and Other SCSI Device Support

OS/2 2.1 supports SCSI-based CD-ROM drives and other SCSI-attached devices, such as read/write optical drives. This support is provided through a combination of the hardware-specific SCSI adapter device driver (.ADD), the hardware-independent SCSI device driver (OS2SCSI.DMD), and a device-specific SCSI peripheral device driver (OS2CDROM.DMD or similar). In addition, a filter device driver can be installed; a typical use of this driver is to convert SCSI-2 commands issued by the OS2CDROM.DMD driver into SCSI-1 commands, which can be understood by more CD-ROM devices.

This layered device-driver approach enables new SCSI CD-ROM drives and other SCSI peripheral devices to be supported with minimum effort, simply by writing a new adapter device-driver (.ADD) file.

The CD-ROM file system support implemented in OS/2 2.1 with CDFS.IFS includes support for CD-XA. OS/2 and DOS applications can thus take advantage of CD-XA. DOS support is provided in the MVDMs through the VCDROM.SYS virtual device driver. This implementation of CD-XA support in VCDROM is independent of the MSCDEX enhancements.

Disk Adapter Device Drivers Supplied with OS/2 2.1		
Device	**ISA Device Driver**	**Micro Channel Device Driver**
Diskette	IBM1FLPY.ADD	IBM2FLPY.ADD
Non-SCSI Disks	IBM1S506.ADD	IBM2ADSK.ADD
PS/2 Model 57 Disk Support		IBM2M57.ADD
SCSI Adapter Device Drivers Supplied with OS/2 2.1		
Adapter		**Device Driver**
Adaptec A/C 6260, AHA-1510, 1520, 1522		AHA152X.ADD
Adaptec AHA-1540, 1542		AHA154X.ADD
Adaptec AHA-1640		AHA164X.ADD
Adaptec AHA-1740, 1742, 1744		AHA174X.ADD
DPT PM-2011, PM-2012		DPT20XX.ADD
Future Domain TMC-845, 850, 850IBM, 860, 875, 885		FD8XX.ADD
Future Domain TMC-1650, 1660, 1670, 1680, MCS-600, 700		FD16-700.ADD
Future Domain FD7000EX		FD7000EX.ADD
IBM 16-bit AT Fast SCSI Adapter		FD16-700.ADD
IBM PS/2 SCSI Adapter and SCSI Adapter with Cache		IBM2SCSI.ADD
Default General ISA SCSI Adapter Support		IBMINT13.I13

Figure 5. OS/2 2.1 Disk and SCSI Device Drivers

At installation time, in most cases, OS/2 senses the installed CD-ROM drive, and then loads the necessary .DMD and .FLT files, and includes the necessary BASEDEV= and DEVICE= statements in the CONFIG.SYS.

OS/2 2.1 does not currently include device drivers for proprietary (non-SCSI-attached) CD-ROM devices. However, IBM is working with CD-ROM drive manufacturers to develop these drivers. Because of the pluggable design of OS/2 2.1, these CD-ROM device drivers could be distributed via bulletin boards and easily installed.

OS/2 2.1 adds CD-ROM support for a range of CD-Technology, Hitachi, NEC, Panasonic, Sony, Texel, and Toshiba CD-ROM drives.

Figure 6 shows the SCSI peripheral support in OS/2 2.1.

OS/2 2.1 Enhancements: OS/2 2.0 included CD-ROM support for the IBM CD-ROM drive. OS/2 2.1 adds CD-ROM support for a range of CD-Technology, Hitachi**, NEC**, Panasonic**, Sony**, Texel, and Toshiba** CD-ROM drives.

Some of these drives require an additional filter driver in order to work correctly in OS/2 2.1. These filter drivers are required to support CD-ROM drives that adhere to the SCSI-1 standard, instead of the new SCSI-2 standard. The new filter drivers are used to convert the SCSI-2 commands generated by the

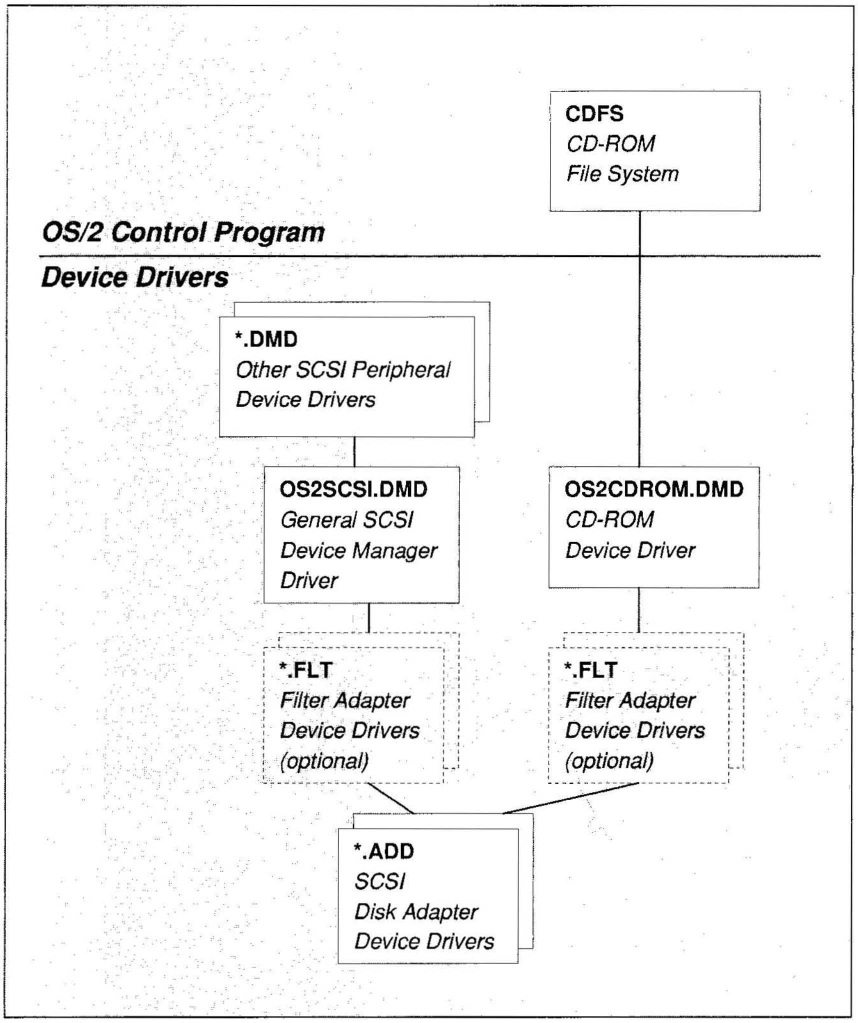


Figure 6. OS/2 2.1 SCSI-Attached CD-ROM and Other SCSI Device Support

OS2CDROM.DMD module into the vendor-unique SCSI-1 commands required by the SCSI-1 CD-ROM drive.

Figure 7 lists the CD-ROM devices supported by OS/2 2.1, along with the corresponding filters found in the \OS2\INSTALL directory. This list is found in the file CDROM.TBL in the \OS2\INSTALL directory, and it is used by the Selective Install program. For all of the devices listed in Figure 7, the device driver is OS2CDROM.DMD.

OS/2 also provides support for the Advanced SCSI Programming Interface (ASPI), developed by Adaptec. This support is provided through OS2ASPI.DMD, which is an ASPI device-manager driver that is compatible with existing device modules written to Adaptec's ASPI specification.

Video Support

OS/2 2.0 was designed to exploit a wide range of display adapters. This support is provided through a range of DLLs and device drivers, even for a single display adapter. This range is necessary to provide video support for a wide range of applications, including DOS, Windows, and OS/2 character-mode applications, as well as native Presentation Manager* applications.

OS/2 2.1 Enhancements: OS/2 2.1 includes the new 32-bit PM graphics engine, and new 32-bit display device drivers for XGA, XGA-2, SVGA, 8514/A, and VGA display adapters.

The SVGA, SGA, SGA-2, and 8514/A display drivers support sessions running in the background and foreground, in full-screen and seamless modes, at resolutions up to 1024 x 768 and up to 256 colors.

Because of its design, OS/2 2.1 is enabled for new display device drivers to be added as they become available from IBM or third parties. IBM is also working with the manufacturers of other leading display adapters and independent software vendors to ensure that the accelerated display functions, such as those implemented on the S3 805 and 926 chipsets, are exploited under OS/2 2.1. These display drivers will be distributed on bulletin boards as they become available.

The SVGA display driver supports a range of SVGA adapters using the following chipsets:

- IBM VGA 256-color
- Cirrus Logic
- Tseng** ET4000
- Western Digital Imaging WD90C11, C30, C31 (C30 mode only)
- Trident Microsystems TVGA8900
- ATI** 28800
- Headland** Technology HT209

CD-ROM Drive	Filter (if needed)
CD Technology T3301	
Hitachi CDR-1650S, 1750S, 3650	HITCDS1.FLT
Hitachi CDR-3750	
IBM CD-ROM I	TOSHCDS1.FLT
IBM CD-ROM II	
NEC Intersect CDR-25, 36, 37, 72, 73, 74, 82, 83, 84	NECCDS1.FLT
NEC MultiSpin CDR-38, 74, 84	
Panasonic CR-501, LK-MC501S, MC501B, MC521	
Pioneer** DRM-600, DRM-604X	
Sony CDU-541, 561, 6211, 7211	
Sony CDU-6111	SONYCDS1.FLT
Texel DM-3021, 5021	SONYCDS1.FLT
Texel DM-3024, 5024	
Toshiba 3201	TOSHCDS1.FLT
Toshiba 3301, 3401	

Figure 7. CD-ROM Device Drivers Supplied with OS/2 2.1

The IBM VGA 256-color chipset is supported only in 640 x 480 x 256 color mode. All the other SVGA chipsets are supported in the following modes:

- 640 x 480 x 256 colors
- 800 x 600 x 256 colors
- 1024 x 768 x 256 colors

New 32-bit display drivers are also provided for VGA adapters (in 640 x 480 x 16 color mode), and XGA adapters (in a range of resolutions up to 1024 x 768 x 256 color mode).

The new 32-bit display drivers cannot be used with the old 16-bit PM graphics engine in OS/2 1.x and OS/2 2.0. The old 16-bit display drivers (such as those for CGA or EGA) still work in conjunction with the 32-bit graphics engine, but may not provide seamless Windows support, or performance as good as that of 32-bit display drivers.

To find out which display drivers are supplied with OS/2 2.1, examine the \OS2\INSTALL directory, and list the files with extension .DSC. These are the display configuration files. Each of these files contains information used by the utility DSPINSTL.EXE to install the video adapter support.

Figure 8 lists the display drivers supplied with OS/2 2.1.

Understanding Video Adapters

Some of the toughest configuration issues revolve around configuring an optimal video solution for OS/2. This section describes the issues involved in video configuration, and ways to navigate successfully through them.

In this section:

- *Video adapter* refers to the video subsystem of a computer, on a board in the bus, or designed into the motherboard; for example, the Orchid** Prodesigner IIs or Catseye XGA-2.
- *Video chipset* refers to a family of chips from one manufacturer that are backward-compatible with each other, and have defining characteristics; for example, Trident 8900A, 8900B, 8900C, or Chips and Technologies 82c451, 82c452, 82c453.

Video is possibly the fastest-changing area of the PC marketplace. Currently, one or more new video chipsets are being introduced each

Mode	Chipset	Resolutions	Display Driver Type	WIN-OS/2 Support
CGA		320 x 200 x 4	16-bit	Full-screen
EGA		640 x 350 x 16	16-bit	Full-screen
VGA		640 x 480 x 16	32-bit	Seamless, full-screen
8514	8514/A	1024 x 768 x 16	32-bit	Seamless, full-screen
SVGA	IBM VGA, 256-color	640 x 480 x 256 (512 KB) 640x480x256 (1MB)	32-bit	Seamless, full-screen
SVGA	Tseng ET4000	640 x 480 x 256 (512 KB) 640 x 480 x 256 (1 MB) 800 x 600 x 256 (1 MB) 1024 x 768 x 256 (1 MB)	32-bit	Seamless, full-screen
SVGA	ATI 28800	640 x 480 x 256 (512 KB) 640 x 480 x 256 (1 MB) 800 x 600 x 256 (1 MB) 1024 x 768 x 256 (1 MB)	32-bit	Seamless, full-screen
SVGA	Cirrus CL-GD5422, CL-GD5424	640 x 480 x 256 (512 KB) 640 x 480 x 256 (1 MB) 800 x 600 x 256 (1 MB) 1024 x 768 x 256 (1 MB)	32-bit	Seamless, full-screen
SVGA	Headland HT209	640 x 480 x 256 (512 KB) 640 x 480 x 256 (1 MB) 800 x 600 x 256 (1 MB) 1024 x 768 x 256 (1 MB)	32-bit	Seamless, full-screen
SVGA	Western Digital WD90C11, WD90C30, WD90C31	640 x 480 x 256 (512 KB) 640 x 480 x 256 (1 MB) 800 x 600 x 256 (1 MB) 1024 x 768 x 256 (1 MB)	32-bit	Seamless, full-screen
SVGA	Trident TVGA8900B, TVGA8900C	640 x 480 x 256 (512 KB) 640 x 480 x 256 (1 MB) 800 x 600 x 256 (1 MB) 1024 x 768 x 256 (1 MB)	32-bit	Seamless, full-screen
XGA	XGA, XGA-2	1024 x 768 x 256 1024 x 768 x 16 640 x 480 x 256 640 x 400 x 256	32-bit	Seamless, full-screen

Figure 8. Display Drivers Supplied with OS/2 2.1

month, along with many boards based on new and old chipsets. Figure 9 shows a graphical representation of some of the competing standards that have been introduced in the last 15 years, and illustrates the lack of standards today in the video marketplace.

This lack of video standards has resulted in the need for new display drivers for each new video adapter that appears. Typically, a manufacturer releases a board once display drivers have been ready for the applications or environments with the main market share; in the past, this has meant DOS (BIOS), Windows,

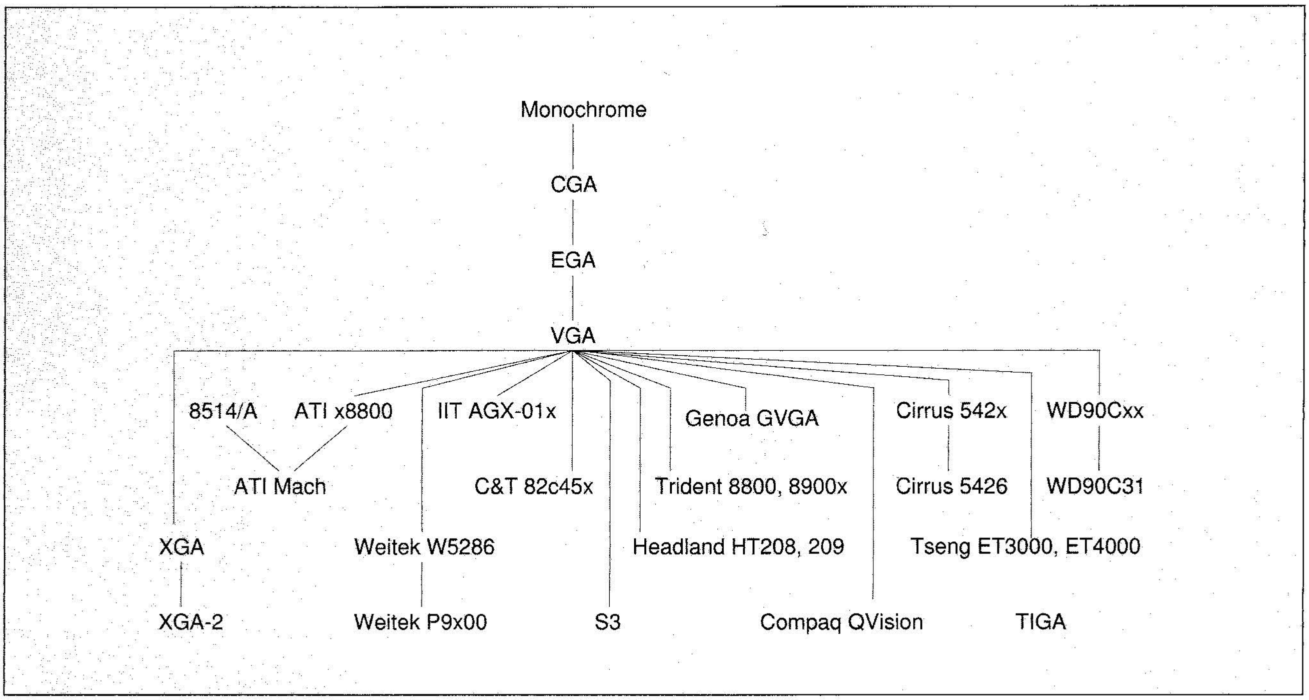


Figure 9. Video Chipset Family Tree

and possibly Autocad. Drivers for older versions of Windows or older revisions of their boards are often not available.

For a new operating system like OS/2 2.x, this has posed significant challenges. Typically, older boards may not be available for sale any longer, but may have a large percentage of the installed base. In addition, developing quality display drivers is a long process. In order to support popular boards, forecasts must be made about which boards will be at their market peak 6 to 8 months later, when the driver development process is complete.

Video Adapter Support and OS/2: OS/2 uses a three-part approach to support the wide range of video boards:

1. The core video standards – CGA, EGA, VGA, 8514, and XGA – are supported.

2. Generic drivers for a large number of popular SVGA chipsets have been developed (Tseng, ATI, CIrrus, Trident, WD, Headland). Manufacturers can then take the source code to these generic drivers, and produce drivers that are optimized for their individual boards.

3. For newer chipsets like S3, IIT AGX, and Weitek P9000, OS/2 drivers are being developed by the manufacturers. S3 has developed a set of OS/2 drivers that support their boards, and source code for 8514 and XGA display drivers is available for manufacturers of boards with similar architectures (such as IIT and Weitek).

Choosing a Video Adapter for OS/2: Choosing the best video adapter for OS/2 is an important consideration. Points to keep in mind include:

- If you are purchasing a video solution for a large number of systems, always standardize on one video-board model for all systems. This will save a great deal of trouble now and in the future.
- Non-accelerated boards (such as Tseng ET4000, Trident 8900/9000, Headland HT209, Cirrus 5424 and lower, and WD 90C30) are much easier to program than are accelerated boards (such as S3, XGA, IIT AGX, Cirrus 5426, ATI Mach, and WD 90C31). Therefore, display drivers for operating systems like OS/2 will be available more quickly for non-accelerated boards. Also, DOS applications rarely use the accelerated functions of the newer accelerated boards, so they may actually be slower for DOS applications.
- When using a graphical user interface like OS/2, however, accelerators take a large burden off the CPU. CPU time that would have been spent moving pixels around

VGA monitor	
640 x 480	60 Hz vertical refresh, non-interlaced
800 x 600	not capable
1024 x 768	not capable
Low-end SVGA	
640 x 480	72 Hz vertical refresh, non-interlaced
800 x 600	56 Hz vertical refresh, non-interlaced
1024 x 768	45 Hz vertical refresh, interlaced
High-end SVGA	
640 x 480	72 Hz vertical refresh, non-interlaced
800 x 600	72 Hz vertical refresh, non-interlaced
1024 x 768	72 Hz vertical refresh, non-interlaced

Figure 10. Typical Video Monitors

on the screen can now be used for applications. This is very advantageous in a multitasking operating system like OS/2, where the CPU can be working on many things at once.

When choosing a particular implementation, it is always best to go with a board that is likely to sell in large quantities, has outstanding software driver support from the manufacturer, and does not use unique or proprietary extensions of the base generic chipset.

Understanding Video Monitors

Computer monitors have different capabilities. VGA monitors can only display 640 x 480 pixels with 60 Hz vertical refresh. SVGA monitors range from displaying 800 x 600 at 56 Hz up to 1280 x 1024 at 72 Hz and beyond. Figure 10 shows profiles of three typical monitors.

Interlaced monitors and lower refresh rates can cause annoying flicker on the screen, which can also be damaging to the eyes. Users will often spend more money to purchase a high-end monitor that does not flicker. Each different monitor model may vary in terms of the refresh rate it supports.

The difficulty arises because there is no standard way for the SVGA video board to detect what kind of monitor is attached to the system. Typically, a video board assumes that the monitor is a low-end SVGA monitor. So, if the user runs a VGA monitor with the board, and attempts to run a high-resolution mode, the monitor goes out of synch. If the person has an SVGA monitor, it will be driven at 45 Hz interlaced at 1024 x 768, regardless of what the monitor is capable of.

Because these SVGA boards were designed for DOS and Windows, the way they handle this problem is to supply a DOS utility in which the user selects the type of monitor attached to the system. Some boards also have Windows drivers hard-coded by refresh rate. This utility then modifies the state of the video board to reflect the type of monitor.

These utilities can be very confusing. It is common to have to run experiments by trial and error, first running the utility, then testing in graphics mode, only to find that the monitor out of synch, which requires rebooting and rerunning the utility to try another setting. Once the utility is configured properly, it is placed in the AUTOEXEC.BAT file or in the CONFIG.SYS file, so it is executed every time the computer is booted. This utility varies from board to board (not just chipset to chipset); each individual SVGA board may have to be programmed differently to enable it to take full advantage of a particular monitor type.

Video Monitors and OS/2: The approach of OS/2 is to use the SVGADATA.PMI file architecture. To properly configure the monitor, the DOS-based utilities are used. These utilities access the video adapter's BIOS to put the board in the proper state, read in the state, store it, and use it for mode sets in protect mode under OS/2.

This sequence is accomplished by using the DOS program SVGA.EXE (by running SVGA ON), which reads the current state of the video adapter, and places it in an ASCII file. The format of this file is based on the VESA protect mode SVGA standard. The file can be edited by a programmer or another program, and is interpreted at boot time and stored in base video data structures. This SVGA.EXE utility is executed, and proper data is generated, before base video will provide any other SVGA services. Because the program attempts to capture the exact state of the adapter at the time it is run, if the user runs

- A status window that shows you the power level of the battery and the state of the battery charge
- A choice to automatically update the status window at user-selectable intervals
- A choice that conserves power by using partial power levels (Suspend function)

Power Status: To display the power status, start the power application from the System Setup folder. If your computer is currently running on battery power, a window similar to Figure 12 is displayed. If your computer is currently running on AC power, the status might look like Figure 13.

The following information is provided in the Power Status window:

1. Power source, displayed as either Battery-powered (operating with a battery pack) or AC-powered (operating with electric current). If the system cannot determine the power source, no Power Source information is displayed.

2. Battery life, displayed as a battery and power-gauge graphic. The power gauge shows the power level of the battery compared to the capacity of the battery. When the shaded area of the gauge is at the top of the scale (100%), the battery is at full power. When it is at the bottom of the scale (0%), the battery is out of power. When the shaded area is dimmed, there is either no battery in the computer, or the computer cannot provide battery information.

3. Battery state. One of the following battery states is displayed:

- High: The battery charge is high, and you can continue using your computer.

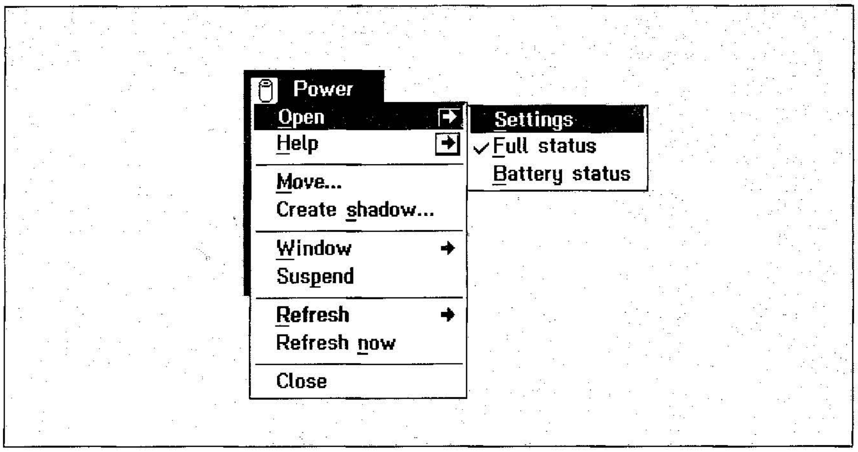


Figure 14. Power - Context Menu

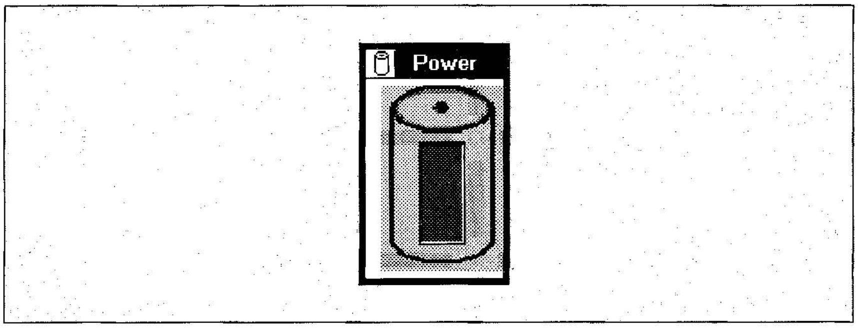
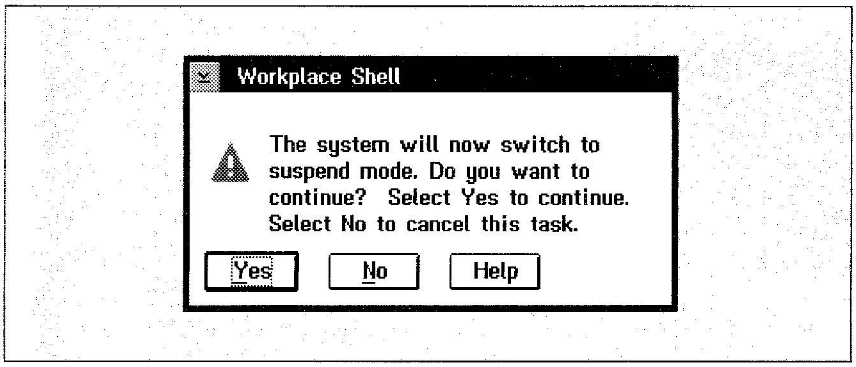


Figure 15: Power - Battery Status

- Low: The battery charge is reduced, and you need to prepare to recharge the battery or switch to another power source, either another battery or AC power.
- Critical: The battery charge is depleted, and you need to recharge the battery or switch to another power source. **Warning:** A critical battery state could cause system failures or data loss. Recharge your battery or switch to another power source *immediately!*
- Charging: The system is restoring the battery charge.
- Unknown: The system cannot determine the battery state.

The context menu of the power application, shown in Figure 14, gives you access to a number of functions:

- Full status or Battery status. Battery status is an alternate selection in place of Full status. Battery status, shown in Figure 15, contains only the battery life information. Due to the reduced size of this view, it is handy to have the battery status always displayed.
- Suspend. The suspend mode allows you to save as much power as possible without turning the computer off. When Suspend is selected, the system dims the display and turns off devices that are not in use. If "Confirm on power status changes" is set in



The system will now switch to suspend mode. Do you want to continue? Select Yes to continue. Select No to cancel this task.

Figure 16. Power - Switch to Suspend Mode

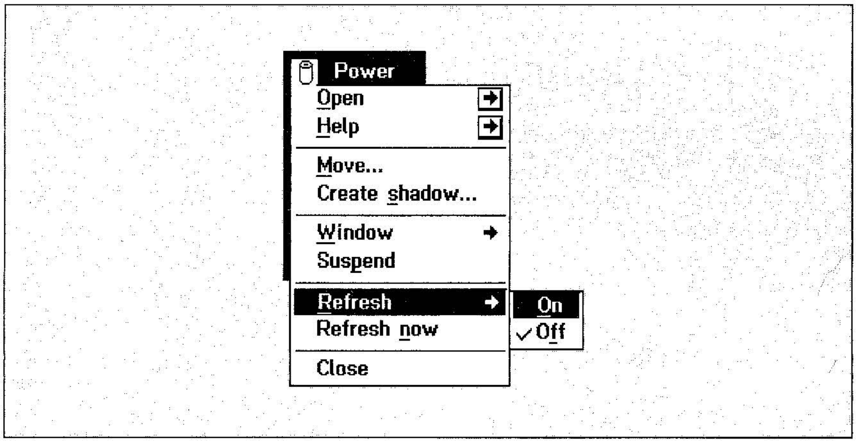


Figure 17. Power - Refresh

the Power Settings, the message in Figure 16 is displayed.

To exit suspend mode and resume operation, you need to use a procedure that differs on different computers. Some systems have a resume key; others enter suspend mode when the computer lid is closed, and exit suspend mode after the lid is opened. Refer to the documentation for your specific computer.

- Refresh. Refresh can be set to On or Off, as shown in Figure 17. If Refresh is set to On, the system automatically updates the status window at intervals that can be set by the user.
- Refresh now. This selection updates the power status immediately.

Power Settings: A number of settings can be chosen by selecting Open - Settings from the status menu of the Power application. The first page of these settings is shown in Figure 18. Here, the power management can be turned on or off.

If "Confirm on power status changes" is checked, the user has to confirm the request to enter suspend mode. (The message shown in Figure 16 is shown in this case.) If "Confirm on power status changes" is not checked, the system immediately enters suspend mode upon request.

The second page of the power settings is shown in Figure 19. Here, the default status view, Full status or Battery status, can be selected.

Automatic refresh can be set to On or Off. If refresh is set to On, the refresh interval can be set from 1 minute to 30 minutes using the spin button.

PCMCIA Support

This section describes the PC Card as defined by the Personal Computer Memory Card International Association (PCMCIA). It also describes the enhancements implemented in OS/2 2.1 to support PC Cards.

A PC Card is a small form-factor adapter for personal computers. It is about the size and shape of a credit card.

Types of PC Cards: PCMCIA standards describe the physical requirements, electrical specifications, and software architecture for PC Cards. Three types of cards are described by the PCMCIA standards:

- Type I Card, 3.3 mm thick, used for various types of memory enhancements, including RAM, flash memory, one-time programmable (OTP) memory, and electronically erasable programmable read-only memory (EPROM).
- Type II Card, 5.0 mm thick, mainly used for I/O features such as modems, LAN adapters, and host connectivity adapters.
- Type III Card, 10.5 mm thick, mostly used as storage devices, such as miniature hard-disk drives.

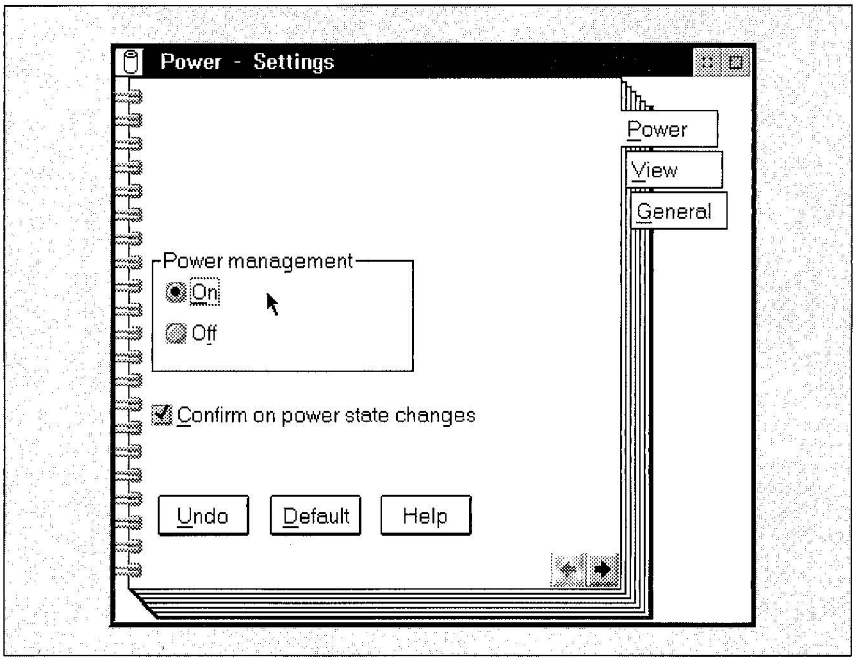
All three card types are 85.6 mm long and 54 mm wide, and they all use the same 68-pin edge connector for attachment to the computer's motherboard.

PC Cards can be used with suitably equipped laptops, notebooks, palmtops, tablets, and other portable computer systems, as well as some desktop computers. PC Cards are a convenient alternative to pocket adapters and docking stations.

Socket Services and Card Services: The key elements of the PCMCIA software architecture are Socket Services and Card Services. Socket Services is a BIOS-level software interface that provides a way to access the PCMCIA sockets (slots) of a computer. Socket Services identifies how many sockets are in a computer system, and detects the insertion and removal of a PC Card adapter while the system is powered on (called *hot-plugging).* Socket Services is part of the PCMCIA 2.0 specification, and it interacts with Card Services.

Card Services is a software management interface that allows you to allocate the system resources (such as memory and interrupts) automatically, once the Socket Services detects that a PC Card has been added. Card Services also releases these resources when the PC Card has been removed. Furthermore, Card Services provides you with an interface to higher-level software to load any needed hardware drivers.

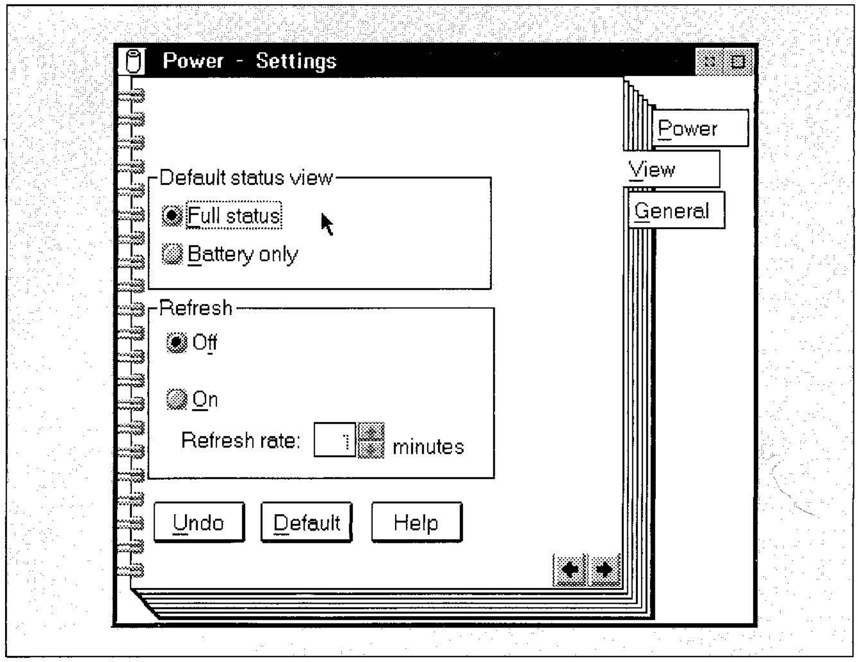
Benefits of PC Card Adapter Technology: The combination of PC Card adapter hardware, Card Services software, and Socket Services software provides a *plug-and-play* capability in the portable computing environment. Once the software has been installed, it is possible to add and remove PC Cards without powering off the system or opening the covers of the computer.



Power management
● On
○ Off

☑ Confirm on power state changes

Figure 18. Power - Settings, Page 1



Default status view
● Full status
○ Battery only

Refresh
● Off
○ On
Refresh rate: 1 minutes

Figure 19. Power - Settings, Page 2

For example, you could insert a modem PC Card to access another computer system, download information into the portable computer's memory, remove the modem PC Card, replace it with a Flash PC Card, and store the downloaded information – all while your portable computer is still powered on.

```
/* */
call RxFuncAdd "SysIni", "RexxUtil", "SysIni"
call SysIni "USER", "PM_IBMVGA", "CURSOR_SIZE", "1",
say Result
exit
```

Figure 20. Manually Configuring the VGA Large Cursor

```
/* */
call RxFuncAdd "SysIni", "RexxUtil", "SysIni"
call SysIni "USER", "PM_IBMVGA", "CURSOR_SIZE", "2",
say Result
exit
```

Figure 21. Restoring the Normal, Small VGA Cursor

As PC Card features are added and removed, you don't have to worry about which memory blocks, I/O ports, or interrupt levels are available. Card and Socket Services software configure the system automatically.

PC Card adapters provide you with the flexibility of adding the features you require after your base system has been purchased. One of the goals of PCMCIA is the interchangeability of PC Cards between portable computers. PCMCIA 2.0 slots can be added to ISA, EISA, and Micro Channel computers. The same PC Card can operate in all these machine types with the appropriate Card and Socket Services software.

OS/2 2.1 PCMCIA Support: OS/2 2.1 provides support for PC Cards that conform to the PCMCIA standard.

If PCMCIA Support is selected during initial OS/2 2.1 installation, or during Selective Install, the two device drivers for PCMCIA support are installed, and the following two lines are added to the CONFIG.SYS:

```
DEVICE=C:\OS2\PCMCIA.SYS
DEVICE=C:\OS2\MDOS\VPCMCIA.SYS
```

These device drivers provide a PCMCIA Card Services interface.

There is no user-visible interface provided with the OS/2 2.1 PCMCIA support. Device drivers and applications to set up PCMCIA cards have to be provided by the manufacturer of the PC Cards. These will include an interface that allows you to set up I/O addresses, RAM and ROM addresses, and IRQ resources used by a PC Card, as required by the specific card you are using.

PCMCIA Compatibility Issues: Some PC Cards were offered before publication of the November 1992 standards, which covered PC Cards, Socket Services, and Card Services. Consequently, some client device drivers may not take full advantage of Card Services and Socket Services, by (for example) not offering hot-plugging. Some manufacturers also bypassed the interfaces, or wrote device drivers directly to applications. It is therefore essential to test the PC, the PC Cards, and the software to ensure compatibility.

VGA Large Cursor Support for LCD Screens

The 32-bit VGA display driver includes a large cursor for use with LCD screens, in order to increase visibility. LCD screens are typically used on laptop systems, such as the IBM ThinkPad series. Large cursors are provided for both the arrow pointer and the I-beam cursor.

The large cursor is not always configured automatically. It can be manually configured by using the REXX commands shown in Figure 20.

OS/2 2.1 must then be shut down and restarted in order to display the new cursor.

The normal, small VGA cursor can be reconfigured by using the REXX commands in Figure 21, then shutting down and restarting the computer.

TrackPoint II Support: The ThinkPad series of notebook computers features an in-keyboard pointing device called the TrackPoint II. Red in color, it looks like a joystick, but it also includes complex algorithms that enable the Trackpoint II to be used in place of a mouse for graphical environments. ThinkPad keyboards also include two extra buttons that correspond to mouse buttons.

OS/2 Device Drivers Available on IBM Personal Computer Company BBS

Since the *OS/2 2.1 Technical Update* was published, many new OS/2 drivers have become available. Below is the list of OS/2 device drivers available from the IBM Personal Computer Company BBS as of mid-January. The phone number for this BBS is 1-919-517-0001. Connect rates range from 1200 bits per second to 14,400 BPS. The line protocol is 8 data bits, no parity, 1 stop bit (8,N,1).

	Size in	Latest	
Driver Name	Bytes	Update	Description
4mmos2.com	17753	01-27-93	2GB Tape Drive driver under SyTos+
rodnt100.zip	62912	07-13-93	3 button mouse drivers-shareware
svga16.zip	1104046	06-29-93	32-bit 800x600 16-color SVGA drivers
os2bt3.zip	619102	07-20-93	ATI graphic cards drivers
macpaopt.dsk	327065	10-29-93	Audio Capture/Playback driver
bt-os2.zip	10240	11-17-93	Buslogic OS/2 2.x SCSI driver
cd-rom2.exe	161812	02-16-93	CD-ROM 2 Option/Driver (DOS,OS/2) v2.02
sio126.zip	141312	01-20-94	COM driver replacements for OS/2
videodr5.zip	57344	01-20-94	Change/install WIN-OS/2 FS drivers
chinon.zip	6381	12-01-93	Chinon 431, 435, 535 CD-ROM drivers
vpros21.exe	1096596	10-26-93	Diamond Viper VLB OS/2 2.1 Drivers
epson.zip	82259	08-22-93	Epson printer driver
hpdjet.zip	96420	08-11-93	HP Deskjet family drivers for OS/2
cdrom.exe	159003	07-16-93	IBM SCSI CD-ROM device driver ver 1.1
iaaos2.dsk	642421	07-29-93	Image-I Adapter/A driver V2.00 - OS/2
ljet.zip	266025	08-11-93	Laserjet family driver
prnfix.txt	8332	08-11-93	List of printer drivers for OS/2
tmv1scsi.zip	53141	12-02-93	MediaVision CD-ROM driver for OS/2 2.1
tmv1scsi.exe	68557	07-26-93	MediaVision beta SCSI drivers
87os221.zip	587776	07-21-93	OAK-087 Prostar SVGA Drivers for OS/2
sdlcdd.zip	13874	11-03-93	OS/2 Extended Services SDLC driver
plotr2.zip	90014	08-11-93	Plotter 2 device driver
post16.zip	206075	08-11-93	PostScript 16 printer driver
post32.zip	196932	08-11-93	PostScript 32 printer driver
sio120.zip	80754	10-23-93	Ray Gwinn's Comm drivers for OS/2, V1.2
sio124.zip	129531	12-22-93	Ray Gwinn's Comm drivers for OS/2, v1.24
s3805d1.zip	993971	09-08-93	S3 driver for OS/2 2.00.1 on VP2, 1 of 2
s3805d2.zip	751612	09-08-93	S3 driver for OS/2 2.00.1 on VP2, 2 of 2
s3-64k.dsk	109897	09-02-93	S3 drivers for OS/2 2.1 (64K colors)
s3-16m.dsk	1420841	01-05-94	S3 driver for OS/2, 16M colors
sony53.zip	13855	01-06-94	Sony CDU 531/535 driver
sony31.zip	24160	01-01-93	Sony CDU-31A drivers for OS/2 2.x
cdu535.zip	5028	09-15-93	Sony CDU-535 CD-ROM driver
sbos2-ne.zip	450964	04-12-93	Sound Blaster driver
34f-os2.zip	56414	11-17-93	Ultrastor 34f SCSI control OS/2 driver
os2-tsl5.zip	75631	09-14-93	TRANTOR SCSI card drivers

Pointing Devices and OS/2 2.x

Craig Eubanks
Additional Technical Support, Inc.
Boca Raton, Florida

Gordon Webster
IBM Corporation
Boca Raton, Florida

Based on information gathered by OS/2 technical support personnel, this article attempts compile all the information related to pointing devices and the various versions of OS/2 2.x, including 2.0, ServicePak 1, and 2.1. This article should be of use to technical support personnel, as well as users attempting to solve their own problems.

There are both technical sections on device drivers and certain types of OEM mice, as well as a common problem/answer section for quick reference while working with customer problems.

All references to C: in this article assume that OS/2 is installed on the C: partition. If this is not true of your computer, simply change C: to the drive letter of the partition in which OS/2 has been installed.

Mouse Device Drivers

The OS/2 operating system provides pointing support for the following:

- IBM 8516* Touch Display
- Microsoft three-byte protocol devices
- PC Mouse Systems five-byte protocol devices

OS/2 2.x has two basic device drivers for pointing devices: a *Physical Device Driver* (PDD) called MOUSE.SYS, and a *Virtual Mouse Driver* (VMD) called VMOUSE.SYS.

There are also two other device drivers provided with OS/2: POINTDD.SYS and PCLOGIC.SYS. POINTDD.SYS (Pointing Device Draw) provides pointer support for OS/2 in the OS/2 full-screen mode only; it is not used in DOS full-screen mode. PCLOGIC.SYS provides support for PC Mouse Systems' five-byte protocol on serial ports only.

Physical Device Drivers for OS/2 2.x: Two classes of pointing devices are supported: *relative* and *absolute.* A relative pointing device reports relative motion – how far the device has moved. An example of a relative pointing device is a mouse, or a trackball. An absolute pointing device reports absolute positions within some predefined work space. An example of an absolute pointing device is a touch-sensitive screen.

Figure 1 contains some commonly used pointing device terms and their definitions.

Generic Pointing-Device Support for OS/2 2.x: The OS/2 operating system provides a physical mouse device driver, called MOUSE.SYS, that attempts to detect the type of pointing device currently installed on the system, as long as the pointing device is 100% Microsoft-compatible (that is, it handles three-byte packets of data). Once MOUSE.SYS detects the existence of a particular pointing device, it dynamically sets up support for that device. The search order for a pointing device is:

1. Pointing Device Interface (PDI) port

2. Serial ports: COM1, COM2, COM3, and COM4

3. Inport (AT bus only)

4. Bus card (AT bus only)

If the physical mouse device driver is unable to detect the presence of a

Term	Definition
PCLOGIC$	The OS/2 system-provided pointing device driver name, which is defined in the device header field of PCLOGIC.SYS.
IDC	Inter-Device Communication. This is the means for communicating between MOUSE.SYS and other mouse device drivers.
Device-Independent Device Driver	Another way of referring to MOUSE.SYS, which handles all the IDC interfaces for pointing devices, and redirects those commands to the rest of OS/2.
Device-Dependent Device Driver	Hardware-specific device driver that reads mouse events and communicates with MOUSE.SYS through the IDC for additional pointing-device support.

Figure 1. Common Pointing Devices and Definitions

Acronyms

COM	Serial COMmunications port, numbered from 1 to 4
DDK	Device Driver Kit
GA	General Availability, a term to describe an IBM product that has become generally available to the public
IDC	Inter-Device Communication
OEM	Original Equipment Manufacturer. Within IBM, this refers to equipment manufacturers other than IBM
PDD	Physical Device Driver
PDI	Pointing Device Interface port, a term often used to describe the dedicated mouse port
PM	Presentation Manager
VDD	Virtual Device Driver
VDM	Virtual DOS Machine
VMB	Virtual Machine Boot

pointing device at installation time, the installation program will prompt the user for pointing device information. The installation program then sets the appropriate statement for the pointing device support in the CONFIG.SYS file. The physical mouse driver will be set up to support the first pointing device that it finds, in case there are two pointing devices on the system.

High-Level Design: During device driver initialization, the physical mouse device driver first checks to see whether the TYPE= overrider has been used. If the DEVICE=C:\OS2\MOUSE.SYS line in the CONFIG.SYS contains a TYPE= overrider, then pointing device support is established through an IDC interface with the device-dependent device driver, using the name that follows TYPE=. The device-dependent device driver must be loaded before MOUSE.SYS, and it must be placed above MOUSE.SYS in the CONFIG.SYS file.

If a TYPE= overrider has not been specified, it is assumed that generic pointing device support is desired.

Physical Mouse Device Driver Considerations: System installation ensures that physical mouse device driver initialization takes place prior to physical asynch (COM port) device driver initialization. This enables the physical asynch device driver to recognize that it is not responsible for servicing the port on which the mouse is installed. In turn, this step ensures that physical mouse device drivers are not preempted from the COM ports by the physical asynch device drivers.

Note: When manually changing CONFIG.SYS, the user must place the MOUSE DEVICE= statements before ASYNC DEVICE= statements (COM.SYS and VCOM.SYS).

Adding Support for a Unique OEM Pointing Device: OS/2 provides a method for supporting additional pointing devices. Pointing-device support can be obtained by writing a device-dependent device driver for the device. This physical device driver communicates with the OS/2-provided, device-independent device driver through the IDC interfaces.

Virtual Mouse Driver: The Intel 80386 processor has a feature that allows a DOS program to run in its own 1 MB address space. This address space effectively isolates the DOS program from the rest of the programs running on the system. This special mode is called *virtual 8086* mode.

OS/2 uses virtual 8086 mode to run DOS applications in their own memory partitions, which are known as Virtual DOS Machines (VDMs). OS/2 can support a large number of these VDMs at one time. Each DOS program runs in its own VDM without any awareness of other programs running on the computer.

DOS programs that write directly to the system hardware or devices are permitted to run in a DOS session. When the program writes directly to a device or the hardware, the operation is trapped by the kernel, and is routed to a *Virtual Device Driver*

(VDD). The VDD is a special type of driver that emulates the functions of a particular hardware device, such as a mouse or COM port. The VDD appears as the actual device to the application, but direct access to the device is, in reality, performed through a Physical Device Driver (PDD), such as MOUSE.SYS. The MOUSE.SYS PDD reads from and writes to the device, and passes the results to the VDD. The VDD then sends the results to the DOS application.

At system boot time, VDDs are loaded after all PDDs, but before the Presentation Manager shell is started. The VDD will not load if the associated PDD is not loaded. In the case of devices, if MOUSE.SYS does not find a pointing device on the system, it will not load itself, and therefore the virtual mouse driver VMOUSE.SYS will also not be loaded. This results in the "SYS1201 VMOUSE.SYS not loaded" error.

When a DOS session is exited, its VDD must perform any cleanup that is necessary, which usually includes releasing any allocated memory, and restoring the state of the device, in this case the mouse.

In OS/2, DOS applications that require the use of a pointing device are supported via the INT 33h interface. There are no restrictions on use of the INT 33h interface, even when a DOS session is in background mode. For example, this interface performs the following functions (among others):

- Position and button tracking and notification
- Selectable pixel and mickey mappings (a *mickey* is 1/80 of a centimeter)
- Pointer location and shape
- Video mode tracking
- Emulation of a light pen

MOUSE.SYS is aware of which session currently owns the pointing device. Thus, when a DOS full-screen session owns the pointing device, MOUSE.SYS notifies the virtual device driver of mouse-type events. In the case of a DOS window, MOUSE.SYS routes events through the Presentation Manager, which in turn routes them to the virtual mouse driver. The DOS setting Mouse Exclusive Access can be set to "on" for the DOS windowed sessions, which then bypasses the Presentation Manager, causing mouse events to be sent directly from MOUSE.SYS to the virtual mouse driver. This option is useful for applications that draw and track their own pointing device, and it cures the problem of having two pointers (arrows) show on the screen in a DOS window.

Virtual Touch Device Driver: The Virtual Touch Device Driver (VTOUCH) provides support for INT 7FH for multiple DOS sessions. By default, this VDD is limited to making actual touch XYZ data available only to full-screen DOS programs (because the PDD, which handles the touch data interrupts, cannot determine which window to send the touch to when running with the Presentation Manager session in the foreground). The physical mouse device driver can determine which window to send the mouse data to, because it is able to feed the single queue of the Presentation Manager, which can then determine which window is to receive the event. If the window is a DOS window, it calls the virtual device driver.

Installation Process

There are important differences between OS/2 2.0 and OS/2 2.1 in the installation process for pointing devices. The changes were made in an attempt to reduce the confusion that caused users to override the system's choice of a mouse driver, and resulted in adding incorrect statements to the CONFIG.SYS file.

OS/2 Version 2.0: In OS/2 2.0, during the installation process, the mouse physical device driver, MOUSE.SYS, attempts to detect a pointing device. If it detects a device, you will *not* be shown a mouse selection panel during the processing of diskette 2.

There are important differences between OS/2 2.0 and OS/2 2.1 in the installation process for pointing devices.

After completing installation of OS/2 2.0, and after the computer is rebooted, it is highly recommended that you do *not* change anything in the PM mouse panel. Many customers are selecting a mouse from this panel, and are forcing a different device type than the one detected.

A prime example of this is the Logitech ** Series-M mouse. Because this mouse is Microsoft-compatible, it will be detected automatically, but the mouse panel will read MS Serial Mouse. The panel also has a choice for Logitech, but it does not state that the choice is only for non-Microsoft-compatible versions. Seeing "MS Serial Mouse" where they expect to see "Logitech," many users then choose Logitech Serial. Because this device is not Microsoft-compatible, making this choice adds the PCLOGIC.SYS driver, which does *not* work for the Logitech Series-M mouse. Thus, users who have Logitech Series-M mice ended up with an incorrect installation if they chose the Logitech selection.

Therefore, if you are not prompted for any mouse information, or if you have a mouse pointer during the first part of the installation process, do not change the settings in the graphical mouse selection panel (i.e., if it ain't broke, don't fix it).

Also note that selective installation has been known to leave undesirable statements behind. For example, if the user originally chose a Logitech mouse in error (when in fact the mouse is MS-compatible), and then later did selective installation back to a PS/2-style pointing device (the MS-compatible selection), on occasion the CONFIG.SYS statements

```
DEVICE=C:\OS2\PCLOGIC.SYS
  SERIAL=COMx (where 'x' is 1 or 2)
DEVICE=C:\OS2\MOUSE.SYS
TYPE=PCLOGIC$
```

are not deleted. This causes problems when in fact the user believes everything should be okay. Check the CONFIG.SYS file for the presence of erroneous statements.

OS/2 Version 2.1: The OS/2 2.1 installation process attempts to detect a pointing device on the system, and then displays the choice. To avoid the problem that occurred during OS/2 2.0 installation, OS/2 2.1 displays the pointing device choice on a panel that is different from the one with the available choices. This should eliminate the confusion noted above. If the installation process is unable to detect the pointing device, or if the user wants to see the other available choices, a second panel appears, with the following selections:

- PS/2-Style Pointing Device
- Serial Pointing Device
- Logitech C-Series Serial Mouse
- Logitech M-Series Mouse
- IBM Touch Device
- PC Mouse Systems Mouse
- Other Pointing Device for Mouse Port
- No Pointing Device Support

In this list, the mouse that is currently selected will be indicated by a black dot next to the selection, and a box around the lettering. If this looks correct, choose OK to continue the installation. If you are doing a selective installation, choose Cancel if the selection is already correct, or OK if you made a change.

CONFIG.SYS Statements

For most pointing devices (not including touch screens), only three statements are needed in the CONFIG.SYS file: a full-screen device driver, a system mouse driver, and a virtual mouse driver for the Virtual DOS Machine (VDM) sessions.

Hundreds of pointing devices are available today. Most of them claim to be Microsoft-compatible. If the device is indeed 100% Microsoft-compatible, the OS/2 mouse device driver supports it. The MOUSE.SYS device driver is able to detect Microsoft-compatible mice during the OS/2 installation process, and then the installation program adds the following three lines to CONFIG.SYS:

```
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\MDOS\VMOUSE.SYS
DEVICE=C:\OS2\MOUSE.SYS
```

Some pointing devices cannot be detected by MOUSE.SYS. Although these devices are not automatically detected, they are still supported (with some exceptions). For example, for certain undetectable pointing devices, when the user chooses the Logitech C-Series selection from the mouse installation panel, the OS/2 installation program adds the following lines for mouse support to CONFIG.SYS:

```
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\MDOS\VMOUSE.SYS
DEVICE=C:\OS2\PCLOGIC.SYS
SERIAL=COMx (where 'x' is 1 or 2)
DEVICE=C:\OS2\MOUSE.SYS
TYPE=PCLOGIC$
```

WARNING: Serial pointing device support for COM ports above COM2 is available only on computers that allow interrupt-sharing, such as IBM PS/2 and EISA computers.

Types of Pointing Devices: This section contains the CONFIG.SYS statements necessary for various kinds of pointing devices. It begins by describing CONFIG.SYS statements for devices that use different types of protocols.

For devices that do not need a custom driver, there are basically three sets of statements: Type 1, Type 2, and Type 4. Type 3 is reserved for devices that can use more than one protocol.

Type 1 pointing devices have two buttons and are 100% Microsoft-compatible. They are automatically detected by the MOUSE.SYS device driver, and do not need a device-dependent driver such as PCLOGIC.SYS. Their CONFIG.SYS statements are:

```
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\MDOS\VMOUSE.SYS
DEVICE=C:\OS2\MOUSE.SYS
```

Type 2 pointing devices are Mouse Systems-compatible. They are generally three-button mice, but do not have to be. Many of the older Logitech devices use the same CONFIG.SYS statements. The CONFIG.SYS statements are:

```
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\MDOS\VMOUSE.SYS
DEVICE=C:\OS2\PCLOGIC.SYS
SERIAL=COMx
DEVICE=C:\OS2\MOUSE.SYS
TYPE=PCLOGIC$
(where 'x' is the COM port number.
Either 1 or 2)
```

Note: IBM three-button mice operate the same as a five-byte protocol Logitech mouse.

The CONFIG.SYS statements for Type 3 pointing devices are as follows:

- If the mouse is in two-button or Microsoft mode, use the Type 1 CONFIG.SYS statements.
- If the mouse is in three-button or Mouse Systems mode, use the Type 2 CONFIG.SYS statements.

Type 4 is for IBM-supported touch screens, such as the IBM 8516 Touch Screen, which is supported only on Micro Channel computers. The CONFIG.SYS statements are:

```
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\MDOS\VMOUSE.SYS
DEVICE=C:\OS2\MDOS\VTOUCH.SYS
DEVICE=C:\OS2\PDITOU02.SYS
CODE=C:\TOUC021D.BIN
INIT=C:\TOUCH.INI
DEVICE=C:\OS2\TOUCH.SYS
TYPE=PDITOU$
RUN=C:\OS2\CALIBRAT.EXE -C
C:\CALIBRAT.DAT
DEVICE=C:\OS2\MOUSE.SYS
TYPE=PDIMOU$
```

Description	Type	Model Number	Protocol	Buttons
Clear Mouse	Serial	LT515	Unsupported	2/3
Kidz Mouse	Serial	unknown	Type 1	2
Logitech Mouse	Serial	CA-93-6MD	Type 2	3
Logitech Mouse	Serial	P7-3F	Type 2	3
Mouseman**	Serial	M-MC13-DB9F	Type 1	3
Mouseman	Bus	M-PD13-9MD	Type 1	3
Mouseman Combo	Serial/PDI	M-CJ13	Type 1	3
Mouseman Cordless	Serial	M-RB24 or M-RA12	Type 1	3
Series 2	PS/2	2-7S	Type 1	2
Series 9	PS/2	CE9-6MD	Type 1	3
Series 9	Serial	CC-93-9F	Type 2	3
Trackman** (old)	Serial	T-CA1-9F	Type 2	3
Trackman (new)	Serial	T-CC2-9F	Type 1	3
Trackman Portable	Serial	unknown	Type 1	3

Figure 2. Logitech Mouse Model Numbers and Protocols

Logitech Devices: Logitech mice and trackballs give OS/2 users the most confusion, and lead to numerous installation problems. This section should help resolve some of the confusion.

The CONFIG.SYS statements needed for each type of Logitech pointing device are shown below. It is not absolutely necessary to run the selective installation to change the OS/2 system configuration for the mice. By changing the CONFIG.SYS file, OS/2 can be reconfigured for each type of pointing device.

In every case, there are some statements that do not need to be changed, but still need to exist. These statements are:

```
DEVICE=C:\OS2\MDOS\VMOUSE.SYS
DEVICE=C:\OS2\POINTDD.SYS
```

Also, CONFIG.SYS must contain statements for either Type 1 or Type 2, as explained previously.

The lines that do change are given below for each specific type of Logitech mouse.

- Bus- and PS/2-type mice will add the following statement if it is not already there:

```
DEVICE=C:\OS2\MOUSE.SYS
```

- M-Series Serial Mice:

```
DEVICE=C:\OS2\MOUSE.SYS
```

- C-Series Serial Mice:

```
DEVICE=C:\OS2\PCLOGIC.SYS
  SERIAL=COMx (where x = 1 or 2)
DEVICE=C:\OS2\MOUSE.SYS
TYPE=PCLOGIC$
```

Known Logitech model numbers and their types are listed in Figure 2.

Interrupts and IRQ Settings

There are three general types of interrupts that can occur in a PC:

- Hardware interrupts
- Software interrupts
- Processor exceptions

In OS/2, most of the problems that users encounter with pointing devices are due to conflicting hardware interrupts. An interrupt conflict will often manifest itself as a non-moving mouse pointer on the desktop.

References

Author	Title	Publisher	Date/Pages
Deitel, H.M. Kogan, M.S.	The Design Of OS/2 (32-Bit OS/2)	Addison Wesley	1992
Desmond, M.	Finding the Perfect Touch	PC World	September 1993
Flynn, M.K.	"Trackballs... Choices for the New Generation of Applications"	PC Magazine	August 1990 pp. 211-277
Howard, B.	"Field Mice... Different, Yes. Better, Maybe."	PC Magazine	October 15, 1990 pp. 111-133
IBM	OS/2 2.0 Technical Library Physical Device Driver Reference Version 2.00	International Business Machines Corp.	First Edition, 1992, IBM# 10G6266
IBM	OS/2 2.0 Technical Library Virtual Device Driver Reference Version 2.00	International Business Machines Corp.	First Edition, 1992, IBM# 10G6310
IBM	OS/2 2.0 Tips and Techniques	International Business Machines Corp.	Sept. 1992, IBM# 53G1930
IBM	International Technical Support Centers, OS/2 Version 2.0 Volume 2: DOS and Windows Environment	International Business Machines Corp.	April 1992, IBM# GG24-3731
Lincoln, S. Lusty, S.	"Keyboards, Mice, and Trackballs With The Personal Touch"	PC World	June 1990 pp. 166-170
Mastrianni, S. J.	Writing OS/2 2.0 Device Drivers In C	Van Nostrand Reinhold	1992
Smith, C.	"Input Devices"	PC World	May 1992 pp. 140-147
Schoeller, P.	OS/2 2.x and Serial Communications	IBM Warranty Support Document	May 1993

***Craig Eubanks** provides OS/2 warranty support at IBM Boca Raton, Florida. He is responsible for analysis and debug of potential defects, primarily in the OS/2 pointing device subsystem, and secondarily in the OS/2 serial communications subsystem. Previously, Craig co-wrote the Presentation Manager interface for the Media Player shipped with Multimedia Extensions for OS/2 2.0, and also worked in OS/2 2.0 MMPM/2 system test. He has a BS degree in physics from the University of Florida and an MS in imaging science from the Rochester Institute of Technology, and he teaches martial arts at Florida Atlantic University. Craig can be reached via Internet at CEUBANKS@VNET.IBM.COM, and within IBM at V$ICRAIG@BOCA.*

***Gordon Webster** is an associate programmer on the OS/2 Device-Driver Kit (DDK) team in Boca Raton, Florida. He answers technical questions about I/O device drivers, monitors e-mail systems for DDK-related questions, reviews I/O documentation, and tests the latest device drivers. Previously, Gordon was in OS/2 development, working on the 16-bit keyboard and mouse API set, and the physical and virtual mouse and keyboard components.*

OS/2 Security Enabling

Raymond J. Martin
IBM Corporation
Boca Raton, Florida

This article introduces IBM's efforts regarding OS/2 security, and defines the direction that IBM plans to take in satisfying customer requirements for OS/2 security.

The OS/2 security strategy is based on IBM's published strategy of providing end-to-end protection of applications and information within an organization. It will provide support for emerging standards, such as OSF's Distributed Computing Environment (DCE) and POSIX. To provide customers with total security solutions, IBM is adding function to OS/2 that will afford a greater level of control and protection of the OS/2 environment.

Additionally, we are developing operating system-level support that will enable the installation and use of various IBM and non-IBM security products for OS/2. These products have been specifically designed and developed to satisfy various customer requirements. As an added benefit in the case of Independent Software Vendor (ISV) products, this effort gives ISVs the ability to develop OS/2 products that can be functionally compatible with their DOS/Windows security products.

Customer Environments

Security in OS/2 is crucial to organizations that must provide enterprise-wide security for their information resources and assets.

Customers currently running DOS/Windows systems in an enterprise network environment have identified certain key requirements for security. These requirements include, but are not limited to: identification and authentication, access control, support for single sign-on, trusted program support, management and audit capabilities, data confidentiality, integrity, and non-repudiation.

For a full discussion of these and other security terms and concepts, refer to the book *IBM Security Architecture*, listed in the Reference section at the end of this article.

Security in OS/2 is crucial to organizations that must provide enterprise-wide security for their information resources and assets.

Many of these customers have invested in IBM and non-IBM software security products to help satisfy these requirements. The cost of the products themselves only partially represents the investments they have made. The cost associated with defining organizational policies; the cost of the effort required to establish, build, and maintain user profiles; and the cost of education and training are also significant.

As customers move to OS/2, and add OS/2 workstations to their LAN Systems environments, they want to have at least as much security capability in OS/2 as they have in DOS/Windows. At the same time, they need to protect the investments they have made in money, time, and effort securing their existing environments.

OS/2 Security Strategy

To create a secure OS/2 system, OS/2 security is being designed to enable and support an Installable Security Subsystem (ISS). The ISS may be an ISV product, an IBM product, or a customer-developed application. Figure 1 represents a secure OS/2 system environment, and depicts the placement and interaction of the various components.

A secure OS/2 environment consists of IBM OS/2 Security Support components and additional components not included in the IBM OS/2 Security Support code. The IBM OS/2 Security Support components are:

- The Personal Logon Facility (PLF), which provides a logon-notebook interface for the control of local and remote user logon, and all logon-related functions such as change password, system lock/unlock, and logoff
- The Personal Desktop Facility (PDF), which provides personalization and object-level access control for desktops; and
- The Security Enabling Services (SES), which enable an Installable Security Subsystem to provide identification and authentication, resource access control, single sign-on, and trusted program capabilities to the workstation environment.

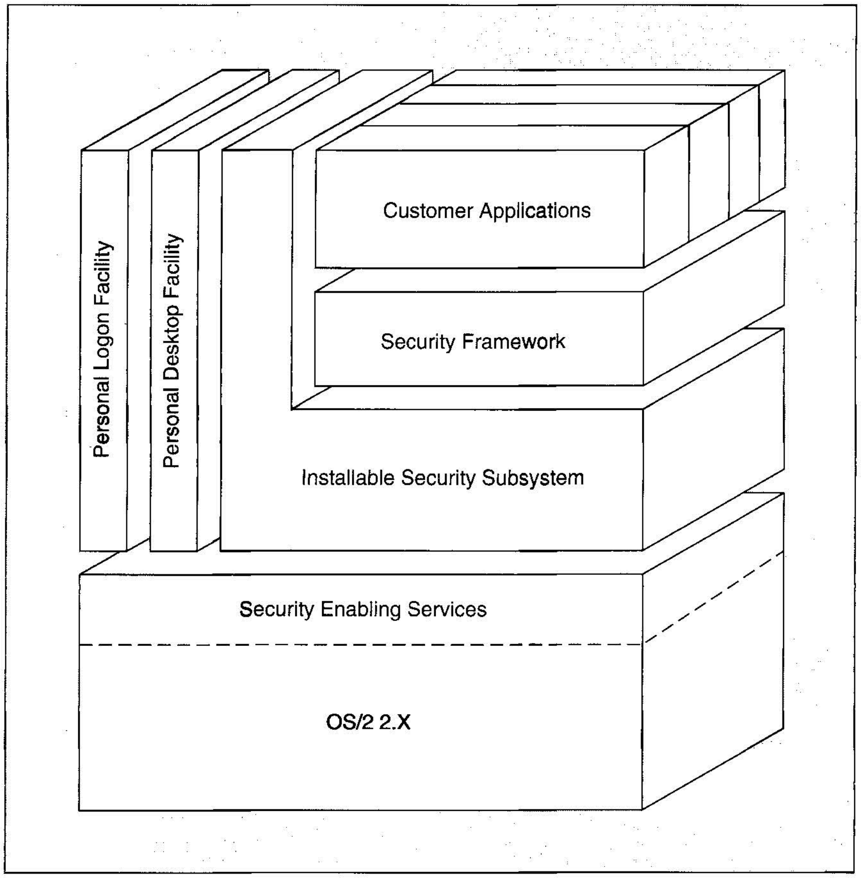

Personal Logon Facility
Personal Desktop Facility
Customer Applications
Security Framework
Installable Security Subsystem
Security Enabling Services
OS/2 2.X

Figure 1. Overview of OS/2 Security Support in a Secure System

The additional components of a secure OS/2 environment are:

- An Installable Security Subsystem (ISS), which is an ISV or IBM product that provides complete security services to the local system and its resources. The ISS is not included in the IBM OS/2 Security Support code.
- A Security Framework, which provides interfaces for applications to communicate to an Installable Security Subsystem. Applications communicate through the Security Framework Application Programming Interface (API), and the framework passes this information down through the Service Provider Interface (SPI) to the ISS. The framework, although identified as a requirement, is currently not planned for inclusion in the initial IBM OS/2 Security Support code.
- Customer (security-dependent) applications, which are the actual programs that exploit the security services of the Security Framework. These applications are not included in the IBM OS/2 Security Support code.

IBM OS/2 Security Support Components

Here are more details about the three major components of IBM OS/2 Security Support.

Personal Logon Facility (PLF): The PLF enables a single logon for local and remote systems. It also provides support for logoff, password changes, system-locking functions, and shutdown. These functions have been designed to interface with, and route information to, a security subsystem that may be an ISV-, IBM-, or customer-developed application.

PLF provides a unique user interface: a logon notebook for storing or selecting information, such as name and password, that is used in performing a single logon. The logon notebook can also be used to specify alternate user authentication mechanisms, such as security cards and badges, voice recognition, and various biometric devices.

Personal Desktop Facility (PDF): Personal desktops may be used on a workstation that supports serial multiple users who have different needs, or are subject to different security policies. A customization function will allow the desktop to be tailored to each user of the workstation, using individual profiles, without affecting the other users. A protection function will set up a customized personal desktop that contains only the object representations of the particular data files and applications that the currently active user is authorized to access. A restriction function provides the ability to impose restrictions on the functions available through the Workplace Shell*.

If the system administrator selects PDF during the OS/2 installation process, then support for desktop customization is automatically included. This gives the desktop manager (the person responsible for customizing individual desktops) the ability to choose the protection level, or the restriction level, as part of the customization process for each user.

PDF also includes rudimentary security services that provide minimal security functions which may be used when a security subsystem is not installed.

PDF works with the desktop protection services, a part of SES, which provides the protection-enforcement mechanism. PDF also requires the security enabling services to assist in the user identification process. This identification information is used to retrieve the current user's personal desktop profile definition from the PDF data base, and to initialize the Workplace Shell.

PDF provides three levels of desktop personalization for the standard Workplace Shell desktop: customization, protection, and restriction. Figure 2 describes each level of personalization, starting with the most basic, and progressing to the most restrictive. Figure 2 also depicts the additive nature of the personalization functions.

Customized Personal Desktop: The Customized Personal Desktop enables more than one user to share an OS/2 system, while allowing each user to customize a personal desktop without affecting other users' desktops. Profiles are maintained so that every time a user accesses the system, the Workplace Shell desktop settings are the same as the user left them. Desktop customization features include color settings, object (icon) positioning, and specification of the particular applications and files to be included on the desktop. The customization level neither prevents a user from modifying or destroying a given desktop configuration, nor prevents a user from using the command line to access applications and files that are not shown on the desktop.

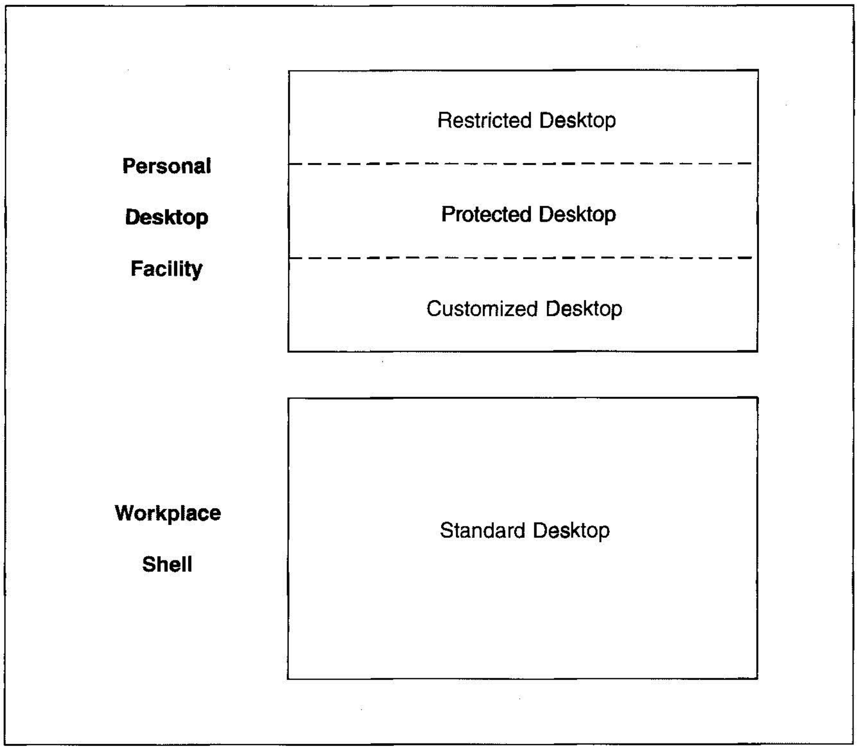


Figure 2. Progressive Levels of Desktop Personalization

Protected Personal Desktop: The Protected Personal Desktop prevents a user from accessing unauthorized files and applications. A user's desktop can be used to implement a protected environment, in which access to files and applications is limited to only those that are defined on the user's desktop by the desktop manager. On the desktop, the user will see only the Workplace Shell objects (icons) that represent the applications and files that the user is allowed to access. The enforcement mechanism monitors requests for access, and enforces the policy while remaining transparent to the end user.

Restricted Personal Desktop: The Restricted Personal Desktop prevents each user from accessing the Workplace Shell features that can circumvent protection policies. Although the desktop manager is able to configure a desktop such that the user cannot access unauthorized data and applications, the desktop manager may want to further restrict users by giving them no access to the tools that could be used to attack the OS/2 system. Some of the predefined and default options of the Workplace Shell menus could be used to attack the protection mechanisms. For this reason, PDF includes a desktop manager's configuration option that enables the manager to redefine and override objects and desktop classes.

Security Enabling Services (SES)

SES provides all the underlying support and operating system interfaces to enable a security subsystem to perform in an OS/2 environment. Assistance is provided in the areas of installation, configuration, and initialization routines; in security event-routing and operating-system services at ring 0; and in providing certain user identification services. This assistance is accomplished through the Security Enabling Services Application Programming Interface (API) and the Kernel Programming Interface (KPI). Some of the services included are:

- Enabling sign-on to the local system, remote systems, and the personal desktop facility (PDF) so that it appears as if it were a single sign-on event
- Providing object-level access control for the protection of objects on a desktop
- Enabling the security subsystem to associate a user with the processes that are executing on behalf of that user
- Routing security-relevant events to the security subsystem, and providing kernel services for the security subsystem
- Enabling the use of alternative, multiple, concurrent authentication devices

Additional Security Components in a Secured OS/2 System

This section expands on the additional security components mentioned above.

Installable Security Subsystem: The Installable Security Subsystem provides the operating system security, including logon identification and authentication, audit, object access control, object reuse protection, administration, and documentation, and other features of a security subsystem. IBM has established an alpha-test program with several security product developers (IBM and non-IBM) to provide early feedback and design validation, as well as to enable them to influence the work being done to OS/2. The intent of this program is to ensure that the correct functions and interfaces are being included.

Security Framework: The Security Framework should be able to provide an API and an SPI for communicating between the security-dependent applications and an Installable Security Subsystem. By using standards-based APIs such as DCE and POSIX, applications could be written independent of the particular security subsystem that is installed.

Customer (Security-Dependent) Applications: Security-dependent applications are those written by the customer to support the business processes, such as teller systems, insurance claims systems, financial systems, personnel systems, and so on.

OS/2 Security Support – Under Development!

OS/2 Security Support provides user interfaces, programming interfaces, and a set of services known as the Security Enabling Services. This support provides the ability to implement single sign-on at the workstation, and offers varying levels of protection of the individual user's desktop environment. In conjunction with this support, an Installable Security Subsystem can provide a complete security solution, giving customers the flexibility to tailor their system security environment to meet their requirements.

Reference

IBM Security Architecture, SC28-8135, offers more details about IBM's stated directions.

***Raymond Martin** is an advisory planner in POS Security Architecture and Enabling within the IBM Personal Software Products division in Boca Raton, Florida. He is responsible for strategy, planning, and marketing support for POS security. Ray joined IBM in 1977 as a systems engineer in the Data Processing Division. Since then, he has held various positions in marketing, management, and development, and has devoted the last seven years to computer and network security. Ray has a bachelor's degree in industrial engineering from Northeastern University, and Master of Engineering Management from George Washington University. His Internet userid is rjmartin@vnet.ibm.com.*

OS/2 2.x Service and Support, Part 1: Installation

In this issue of Personal Software, *we begin carrying installments of the IBM publication* OS/2 2.x and OEM Hardware, *produced by the Worldwide OEM Technical Service and Support group in Boca Raton, Florida. This book was issued by Robert J. Dilena and prepared by Kirk J. Krauss.*

The first installment covers OS/2 installation, offering a collection of details and tips from IBM support personnel.

Installation problems are probably the most common problems faced by new OS/2 users. Many of these problems are hardware-dependent. In fact, any computer that runs OS/2 successfully may be said to be in excellent working order, because the installation of OS/2 is a rigorous test of a computer's memory, hard drive, video circuitry, and adapters. The computer's primary components must work well together, or OS/2 installation will not complete.

OS/2 requires a large commitment of computer resources, and anyone who installs OS/2 should take several factors into account, such as:

- How many partitions do I need on my hard drive? How large should the partitions be?
- Do I need DOS and Windows application support?
- Does my system have any devices that will require third-party device drivers? If so, how will I install them?
- Which file system, FAT or HPFS, should I use?

The Installation Guides that come with OS/2 2.0 and 2.1 offer advice about such concerns. However, you can install OS/2 quite easily by selecting the default settings provided by the installation process. You need only make some choices based on your requirements and your computer's hard drive or RAM space limitations.

Beginning the Installation

Boot the OS/2 2.x Installation Disk. When the system prompts for another disk, insert Disk 1 and press Enter.

If you choose to install OS/2 in a partition other than the default partition, the FDISK utility will run. Using FDISK, you can create and delete drive partitions, select partition names, and set one partition "installable," so that OS/2 will install in that partition. FDISK also lets you install the Boot Manager, a utility that runs every time the computer is booted, and allows you to select the partition containing the operating system that you want to boot.

We recommend that you place OS/2 in a partition that is separate from your applications and data. Also, if you want to run DOS or other operating systems, you should install the Boot Manager.

Boot Manager: To set up the Boot Manager, refer to the *OS/2 2.0 Installation Guide* (IG), page 42, or the *OS/2 2.1 Installation Guide* (IG), page 68. The Boot Manager occupies its own 1 MB partition, which must be placed within the first 1 GB of hard-drive space.

The FDISK utility starts automatically some time after Disk 1 is inserted. FDISK sets up the Boot Manager and the other partitions as follows:

- Choose the Free Space option. A new set of options appears.
- Select Install Boot Manager.
- Specify a place to put the Boot Manager's partition (anywhere within the first 1 GB is fine).

Each time it runs, the Boot Manager reads pointers to the partition boot blocks. The Boot Manager can be deleted or bypassed at a later time, if needed, so that boot problems can be analyzed. Whenever you run FDISK, FDISK looks for the Boot Manager, and it automatically makes the Boot Manager startable, even if the system was previously set up otherwise.

The OS/2 Partition: Make a partition for OS/2 2.x as follows:

- Press Esc to return to the FDISK main menu.
- Choose the Free Space option, as before.
- Select Create Partition, and enter the size in MB.

If the hard drive has 4 MB cylinders, the bootable partition can be as large as 2 GB under FAT, and 4 GB under HPFS.

OS/2 2.x requires at least 40 MB for a complete setup. Use 50 MB or more if you are installing additional products such as Extended Services and LAN Server. For an optimal OS/2 setup that includes these product extensions, use at least 70 MB if possible.

- Specify whether this is a "primary" partition. Make it "logical" if more than two primary partitions will be used for other purposes (see the 2.0 IG, page 50, or the 2.1 IG, page 68). OS/2 2.x will work from a logical partition, whereas DOS and OS/2 1.x will not.
- "Create Partition" will set up the partition, but the new partition will not be added to the Boot Manager menu without the following additional options:
  - "Add to Boot Manager menu" will prompt for a title such as "OS/2 2.1."
  - "Assign C: partition" will assign a logical drive letter for the partition.
  - "Startup values" (see 2.0 IG page 54 or 2.1 IG page 72).
  - "Set installable" (see 2.0 IG page 54 or 2.1 IG page 73).

Add other partitions to the Boot Manager menu as needed. For instance, a separate partition is recommended for applications, to avoid the necessity of reinstalling them in case OS/2 is accidentally corrupted (by, for example, a DOS utility).

If your hard drive has 1 MB cylinders, the bootable partition is limited in size to 1 GB under both the FAT and HPFS file systems. If the hard drive has 4 MB cylinders, the bootable partition can be as large as 2 GB under FAT, and 4 GB under HPFS. In current versions of OS/2, non-bootable partitions can be as large as 2 GB under FAT, and 64 GB under HPFS.

- Press F3 to save and exit FDISK. Continue as prompted.
- Install the first five disks. The system will prompt for each additional disk as needed. Disk 2 asks you to choose FAT or HPFS; make a choice and continue.

While HPFS may provide better performance than FAT for large partitions (150 MB or greater), HPFS may also create complications for certain applications as well as certain computers. FAT is the better choice for computers with small hard drives (less than 150 MB). Do *not* install HPFS on computers that have less than 6 MB of RAM.

After Disk 5, the system prompts for the Installation Disk once again. Continue following the prompts.

Graphical Installation: The Presentation Manager (PM) shell is loaded after the system reboots from the hard drive. Once PM is running, the prompt for Disk 6 appears. After this point, paging is enabled, and a generic mouse driver is provided. A dialog box with radio buttons is displayed, and the following options are provided:

- Learn how to use a mouse
- Install a basic set of preselected features, which requires less hard-drive space than a complete installation
- Install all features of OS/2, and accept the default values for system options and parameters, for a complete installation
- Select particular options and features to install, allowing modification of system parameters

After you select one of these options, a System Configuration menu appears. Choose the following items from list boxes:

- *Mouse:* The type of mouse should be recognized automatically. Modify this setting if no mouse is attached now, but one will be added later.
- *Keyboard:* The type of keyboard should also be recognized automatically.
- *Display:* The video adapter should be recognized. Modify this setting if the computer has a non-standard display setup.
- *Country:* Copies of OS/2 sold in the USA default to USA.

If you have chosen to select features and install (in the first list box), a menu of optional features appears, prompting you for the features you want to include:

- CD-ROM device support (not available in OS/2 2.0)
- On-line documentation
- Fonts
- Optional system utilities
- Tools and games
- Virtual DOS and WIN-OS/2 support
- HPFS
- REXX
- Serial device support
- Serviceability and diagnostic aids
- Optional bitmaps

The size of each option is shown in MB. You can also select an option called Software Configuration, which gives you a menu of parameters that you can alter in the CONFIG.SYS file on the hard drive.

After these installation requirements are selected, installation continues. A progress indicator appears, showing both the name of the file being installed and the percentage of data that has been copied from each disk.

Shortcut If Installation Stops: From this point forward, if the installation procedure stops before completion, you might be able to complete the installation without starting from the beginning. This is done as follows:

- Boot OS/2 from the floppy disk.
- Use an editor to edit the CONFIG.SYS file on the hard drive.
- On the line that starts with FIRSTDISK=, change the number to the number of the disk that was in the drive when installation stopped.
- For the line that starts with NUMDISKS=, subtract the FIRSTDISK= number from the total number of installation disks (not counting printer drivers and display drivers), and add 1 to the result. Put that number here for the number of remaining disks.

Example: For OS/2 2.0 installed from 3.5-inch media, the total number of disks is 15. If installation stops at Disk 7, edit CONFIG.SYS so that it contains these statements:

```
FIRSTDISK=7
NUMDISKS=9
```

The system should preserve any selective installation choices that you have made. See the next major heading, "The Installation Process: Behind the Scenes," for additional guidelines.

- Remove the OS/2 boot disk from the diskette drive.
- Press Ctrl-Alt-Del. After it reboots, the system should request the disk specified in the FIRSTDISK= statement.

Completing the Installation: The number of installation disks varies according to the OS/2 version, the compression method, and the choice of 3.5-inch or 5.25-inch media. After the last disk, you are prompted to install support for your printer. If you have more than one printer, you can install support for the additional printers after OS/2 is up and running.

When installation completes, the system reboots, and the PM desktop is generated. A tutorial screen appears, but wait to use it until the full desktop is constructed.

Once the desktop is created, be sure to shut down the system, and to reboot it, *before* you run any DOS or Windows applications.

Redirected Installation Disks: For OS/2 2.0 and for OS/2 2.1 on salmon-colored diskettes, a modified version of the 5.25-inch Disk 1 lets you install OS/2 on a computer that has a 5.25-inch, 1.2 MB, bootable Drive A, and a 3.5-inch, 1.44 MB Drive B. To install, you will need:

- The provided 5.25-inch OS/2 Installation Disk, which replaces the corresponding 3.5-inch disk
- The provided 5.25-inch OS/2 Disk 1
- All of the original 3.5-inch Installation Disks

To install, follow these steps:

- Place the 5.25-inch Installation Disk in Drive A.
- Place the 3.5-inch Disk 1 in Drive B.
- Boot the system.
- When prompted, place the 5.25-inch modified version of Disk 1 in Drive A, and press Enter.
- Do *not* remove the disk from Drive A until the hard-drive preparation is complete.

The panels that appear during installation will ask you to remove each disk from Drive A and to place the next disk in the drive as needed to complete the installation. At these panels, you should place the requested 3.5-inch disks in Drive B.

```
D:\INSTALL\LOADDSKF D:\OS2\DISKIMGS\35\disk.DSK A:
```

Figure 1. Invoking LOADDSKF

```
D:\INSTALL\LOADDSKF D:\OS2\CDINST\35\DISK0.DSK A:
D:\INSTALL\LOADDSKF D:\OS2\CDINST\35\DISK1.DSK A:
```

Figure 2. Unpacking the Installation Disk and Disk 1 from CD-ROM

Unsupported CD-ROM Drives: If you have a CD-ROM drive that is supported under DOS but not under OS/2, you can still use it to install OS/2, as follows:

- Boot DOS.
- Use the LOADDSKF utility, found in the \INSTALL directory on the CD, to make a set of OS/2 installation disks from images on the CD.
- For each image, place a formatted 3.5-inch disk in Drive A, and invoke LOADDSKF, with a command such as shown in Figure 1, where D: is the CD-ROM drive and `disk` is the name of the OS/2 disk image to be loaded to the 3.5-inch floppy disk in Drive A. There is also a \525 subdirectory containing the 5.25-inch disk images.
- Disconnect the CD-ROM drive and any associated sound card.
- Install OS/2 from the newly created set of installation disks.

The CD also contains images of the CD-ROM-installation version of the Installation Disk and Disk 1. These images are usable for installing the remainder of OS/2 from many CD-ROM drives. They are unpacked from the CD with the commands shown in Figure 2. Again, there is a \525 subdirectory containing the 5.25-inch disk images for this pair of disks.

The following additional steps may provide access to the CD-ROM drive from within OS/2:

- Copy the hard-drive device drivers, which include the .ADD files and IBMINT13.I13, to the system's root directory or \OS2 subdirectory. Be sure to copy the correct device driver for the hard-drive adapter.
- Reconnect the CD-ROM drive (and the sound card, if present).
- Do a Selective Install of the CD-ROM drive.

The Selective Install process should add the following lines to CONFIG.SYS (assuming C: is the drive on which OS/2 resides):

```
DEVICE=C:\OS2\OS2CDROM.DMD /Q
IFS=C:\OS2\CDFS.IFS /Q
BASEDEV=AHA154X.ADD
```

If the computer is still unable to access the CD-ROM drive from within OS/2, it may be because the following files are not present on the hard drive:

```
OS2CDROM.DMD
CDFS.IFS
UCDFS.MSG
UCDFS.DLL
```

You cannot use the Selective Install procedure to install these files, because OS/2 cannot access the CD in order to copy them. OS/2 may not contain the necessary device drivers for the CD-ROM drive; however, the following CD-ROM drives can be supported by copying the generic CD-ROM drivers from Disk 1 of the OS/2 installation set to the computer's hard drive:

- Creative Labs Sound Blaster Pro CD-ROM drive (Panasonic 521)
- Hitachi 1503S, 1700S, 1900S, 3500, 3600, 3700, 6700
- Mitsumi** LU002, LU005
- Sony 31A, 7305

To install support for the Sony, Mitsumi, and Hitachi drives, copy the files OS2CDROM.DMD and CDFS.IFS from Disk 1 to the \OS2 directory on the hard drive. Add the following lines to the end of the CONFIG.SYS file on the hard drive:

```
DEVICE=C:\OS2\OS2CDROM.DMD /Q
IFS=C:\OS2\CDFS.IFS /Q
```

For the Sound Blaster** drive, copy only the CDFS.IFS file, and add only the "IFS=..." line above to CONFIG.SYS.

Once these files are copied, the Selective Install procedure should work. Go to System Setup, choose Selective Install, and select the CD-ROM Device Support check box from the System Configuration menu. Choose OK to get a CD-ROM selection list, and choose Other (at the bottom of the list). Then complete the Selective Install procedure normally. After the computer is rebooted, the CD-ROM drive should be supported.

Selective Install Problems and Data Compression: Under OS/2 2.1, the Selective Install procedure may request the wrong number of disks if you have a different installation set from the one originally used to install OS/2. An error message may also appear, stating the disk is not readable. Such problems occur because two types of compression have been used on OS/2 2.1 installation sets. The blue-labelled installation set (Type 0) uses one type of compression, and the salmon-labelled set (Type 2) uses another, more effective type of compression. A compression compatibility package is available from the IBM National Support Center BBS, the IBM Support Center, and CompuServe.

Running CHKDSK /F on a System Installed from CD-ROM: To run CHKDSK on a system installed from CD-ROM, do the following:

- Boot the Installation Disk, and insert Disk 1 when prompted.
- Place the CD in the CD-ROM drive when prompted.
- At the Welcome screen, press Esc to exit to an OS/2 command prompt.
- Change directories to the \OS2SE2\DISK_2 directory on the CD.
- Enter CHKDSK x: /F, where x: is the drive on which OS/2 is installed.
- If any errors are found, repeat the CHKDSK x: /F command until no further errors are reported.

Installing a Printer Driver: You can add printer support as follows:

- Open the Templates folder, and drag-and-drop a printer template from the folder to the desktop. The "Create a Printer" screen should appear.
- Place a name for the new printer in the Name field.
- If a suitable printer driver already appears in the Printer Driver window, select the driver and output port, and choose Create. A new printer object will be set up.
- If the correct printer driver does not appear, click on the Default printer driver with the right mouse button, then choose Install. A window titled "Install new printer driver" should appear.
- Place Printer Driver Disk 1 in Drive A, ensure that A:\ is listed as the source directory, then select Refresh.
- Scroll through the printer drivers shown. If the correct driver does not appear, try the other Printer Driver Disks by inserting them and selecting Refresh again.
- When the correct printer driver appears, select Install. Once the driver has been installed, Cancel the "Install new printer driver" window, then select an output port and choose Create. A new printer object will be set up.

The system may also prompt for installation of the equivalent WIN-OS/2 printer. If one may be needed, choose Yes and follow the prompts.

Installing a Printer Driver on a Preloaded System: Printer support is added somewhat differently on computers that come preloaded with OS/2 2.0. A printer driver may be added from the Welcome folder (under Configuration Tools). You may also drag a printer template to the desktop, but this procedure is also somewhat different from the method given previously.

To install a printer driver using the Welcome folder, do the following:

- Open the Welcome folder and, once inside, open the Configuration Tools folder.
- Choose Configure.
- Select the box to the left of Printers, and choose OK.
- Select a printer from the list provided, and choose OK.
- Follow the prompts.

To install a printer driver using the Templates folder, do the following:

- Open the Templates folder, and drag-and-drop a printer template from the folder to the desktop. The "Create a Printer" screen should appear.
- Place a name for the new printer in the Name field.
- Click on the Default printer driver with the right mouse button, then choose Install. A window titled "Install new printer driver" should appear.
- In the Directory box, enter the path of the printer drivers. On preloaded systems, the printer drivers are located in paths as shown in Figure 3.
- Select Refresh to bring up a list of printer drivers.
- Scroll through the printer drivers shown. If the correct driver does not appear, try the other Printer Driver Disks by changing the path in the Directory box to another of those shown above.
- When the correct printer driver appears, select Install. Once the driver has been installed, Cancel the "Install new printer driver" window, then select an output port and choose Create. A new printer object will be set up.
- The system may also prompt for installation of the equivalent WIN-OS/2 printer. If one may be needed, choose Yes and follow the prompts.

```
C:\OS2\INSTALL\PRTDRVS\PMDD_1  (Printer Driver Disk 1)
C:\OS2\INSTALL\PRTDRVS\PMDD_2  (Printer Driver Disk 2)
C:\OS2\INSTALL\PRTDRVS\PMDD_3  (Printer Driver Disk 3)
C:\OS2\INSTALL\PRTDRVS\PMDD_4  (Printer Driver Disk 4)
C:\OS2\INSTALL\PRTDRVS\PMDD_5  (Printer Driver Disk 5)
```

Figure 3. Printer Driver Paths on Preloaded Systems

The Installation Process: Behind the Scenes

Depending on the type of installation you select, over 30 MB of files may be installed during the OS/2 installation process. As each file is installed, its name is added to the \OS2\INSTALL\INSTALL.LOG file. Error messages are also logged there. Some files are copied directly from the floppy disk to the hard drive, but most files are unpacked from a compressed BUNDLE file, and a few files are generated by the installation process to match user and system specifications.

The system is booted a total of four times during the OS/2 installation process. At each boot, three hidden files are processed, the system enters protect mode, and the OS/2 kernel establishes control of the system. Most known installation problems are really boot problems or Workplace Shell (WPS) problems, rather than problems in the installation process itself.

OS2BOOT, OS2LDR, OS2KRNLI Files: The Installation Disk contains the three files necessary to boot OS/2 and load the operating system kernel. The OS2BOOT file loads the loader (OS2LDR), which in turn loads the kernel (OS2KRNLI). When the IBM logo screen appears, the OS/2 kernel is operating, and the processor is in protect mode. The kernel file displays this logo screen when its filename is still OS2KRNLI. It is renamed to OS2KRNL when it is copied to the hard drive during the second pass of Disk 1.

The First File System: A small, generic file system is loaded when Disk 1 is processed. An INT13 device driver is loaded for hard-drive access.

The first CONFIG.SYS File: The installation facility tests the system to determine which drivers it will need in order to begin installing OS/2 on the hard drive. For instance, if the computer is a Micro Channel system, this fact is detected here.

A default version of the CONFIG.SYS file is present on Disk 1. The CONFIG.SYS file establishes system configuration. The OS/2 boot sequence reads this file to determine OS/2 system parameters.

If the CONFIG.SYS file contains the line PROTSHELL=CMD.EXE, an OS/2 command prompt is displayed when the system is booted.

FDISK and COUNTRY.SYS: If the installation facility failed to recognize the hard drive, FDISK will not be able to write to the drive. FDISK is the first routine that attempts to access the drive.

If you choose to install without repartitioning, COUNTRY.SYS is the first file read from the hard drive.

Again - OS2BOOT, OS2LDR, and OS2KRNLI: After the OS/2 base and PM files have been unpacked from Disks 2 through 5 onto the hard drive, you are asked to insert the Installation Disk again. The OS2LDR file is then copied to the root directory of the hard drive as a system, hidden, read-only file.

The OS/2 kernel is also copied as a system, hidden, read-only file. Its filename is changed from OS2KRNLI (on the Installation Disk) to OS2KRNL (on the hard drive), so that the IBM logo screen does not appear when OS/2 is booted from the hard drive.

The OS2BOOT file is created, not copied, because it is specific to the computer and to the partition where OS/2 is installed. This file also has system, hidden, and read-only attributes.

Second CONFIG.SYS File: A new copy of CONFIG.SYS is created on the hard drive. Parameters defined in this CONFIG.SYS file are based on system details that were automatically recognized by the installation routine.

Second Boot: The computer boots from the hard drive using the driver chosen by the installation routine. The Presentation Manager, Workplace Shell, and final file system are now operational.

Installation problems may occur at this point, because this is where OS/2 boots from the hard drive for the first time, and this is also where the Workplace Shell's video support is first used.

The OS2BOOT file is created, not copied, because it is specific to the computer and to the partition where OS/2 is installed.

Note: Any changes made to the CONFIG.SYS file on Disk 1 should also be made here, particularly if a workaround was used at Disk 1 to get past a problem that recurs either here or during the processing of Disk 6.

MAKEINI: At Disk 8, a new thread is created to make the .INI files. These files, and to some extent the EA files, control the icon arrangements, colors, and general "feel" of the PM desktop.

If the system hangs after Disk 8, the .INI files may not yet be complete. Going back for a moment, recall the shortcut you can take if installation stops. In this case, it is safest to set FIRSTDISK=8 if the installation hangs at any point between Disk 8 and Disk 11.

Some systems may hang at Disk 9 because of a timing problem. A new version of UNPACK.EXE has been created as a possible solution to this problem.

MIGRATE.EXE: DOS and Windows programs can be migrated to the OS/2 desktop. Migration is simply the creation of a desktop icon for a program that is present. Migration of any DOS CONFIG.SYS and AUTOEXEC.BAT files that may be present is not recommended. If Windows 3.1 is installed on the computer, and you are installing OS/2 2.0, do not migrate the Windows desktop, because it will corrupt WIN-OS/2.

Differences Between OS/2 2.0 and 2.1 Installations: Under OS/2 2.0, the system asks you to specify a printer driver after installation has completed through Disk 15. In OS/2 2.1, this decision is made at Disk 6, when you set the other installation parameters. Also, OS/2 2.1 asks you to specify CD-ROM support.

Third CONFIG.SYS File: The copy of CONFIG.SYS on the hard drive is now updated with settings that you have chosen since the last reboot, the one that occurred before graphical installation began at Disk 6. If a crash occurs after this point, the FIRSTDISK= shortcut no longer works.

Third Boot: The desktop is built after the system boots again. There is no message stating that the desktop is being built.

Sometimes, if you are reading the tutorial, the build of the desktop hangs. There is no warning when this occurs, but the desktop will not be built. In this event, boot OS/2 from floppy disks and rebuild the

.INI files using MAKEINI, or reboot and press Alt-F1 several times.

CALL Statements in CONFIG.SYS: The CONFIG.SYS file can include a command such as that shown in Figure 4.

```
CALL=C:\OS2\XCOPY.EXE C:filename D:filename
```

Figure 4. Sample CALL Statement in CONFIG.SYS

You can protect your .INI files by having them backed up each time you boot OS/2. The two lines in Figure 5 will back up the current .INI files, as well as the last backup (assuming that OS/2 is installed on Drive C):

```
CALL=C:\OS2\XCOPY.EXE C:\OS2\*.INX C:\OS2\*.INY
CALL=C:\OS2\XCOPY.EXE C:\OS2\*.INI C:\OS2\*.INX
```

Figure 5. Backup CALL Statements

Fourth Boot: Once the desktop has been built, be sure to shut down and reboot again before proceeding to install or run applications.

Booting OS/2

While you are installing OS/2, your system is booted four times. In fact, most perceived installation problems would be better described as boot problems or driver compatibility problems, because the installation process itself cannot start until OS/2 has been booted and the computer's hard drive and video adapter have been recognized. Once it has been successfully installed, most users will continue to boot OS/2 again and again. Each time, the sequence of events will be much the same as it was during the initial installation of OS/2.

During boot, from the time the computer's power is turned on until the OS/2 desktop appears, control of the system passes through several procedures that can be roughly described as follows:

- Power-On Self Test (POST)
- Processing of the OS2BOOT file
- Processing of the OS2LDR file
- Processing of the OS2KRNL file
- Processing of the CONFIG.SYS file
- The Workplace Shell becomes operational

Creating a Bootable OS/2 2.x Disk: A bootable OS/2 disk is useful for several purposes, such as accessing an HPFS drive quickly, or checking the CONFIG.SYS file. OS/2 2.0 is recommended for this purpose, because the OS2KRNL file in OS/2 2.0 is much smaller than in OS/2 2.1, and a bootable disk requires additional space for some other files. A utility called BOOT21D, available from some BBSs, can be used to make a bootable OS/2 2.1 disk. Another utility, MKBTDSK.ZIP, automatically creates a boot disk using the OS/2 2.0 Installation Disk and Disk 1. This utility is available on CompuServe, the IBM PC Company BBS, and on other OS/2 BBSs.

You can also create your own boot disk by using the SYSINSTX utility to place the necessary OS/2 files onto a fresh floppy disk, as follows.

The SYSINSTX command is similar to the DOS SYS command. It creates the OS2BOOT hidden file on a specified partition or drive. To use SYSINSTX in an HPFS partition, you must have UHPFS.DLL in the path, along with SYSINSTX.COM.

Copy SYSINSTX.COM from the Installation Disk to the OS/2 partition, which is Drive C. Now, put a formatted 1.44 MB diskette into Drive A, and type SYSINSTX A:

The other three OS/2 files on the installation disk are not hidden, so you can use the COPY command to copy them from Drive C to Drive A. Don't forget to change name of OS2KRNLI to OS2KRNL.

The other required files are on Disk 1. Files with names containing the numeral 1 (such as IBM1FLPY.ADD) are for ISA and EISA systems; files with the numeral 2 are for Micro Channel systems.

On Disk 1, there is a file named OS2SCSI.DMD which is not listed below. This device manager is often needed for SCSI drives, and it requires a corresponding CONFIG.SYS statement if it is included.

For ISA systems with non-SCSI drives, the files in Figure 6 provide a basic but usable OS/2 2.0 on a high-density floppy disk.

If PRINT01.SYS were not included, the remaining files would be able to fit on a 1.2 MB 5.25-inch disk. On a 3.5-inch disk there is sufficient space for CHKDSK.COM and a modest-sized editor. Even with these additions, some space is left over.

The CONFIG.SYS File on the Bootable OS/2 2.0 Disk: Be sure to include the following lines in the CONFIG.SYS file on the bootable OS/2 disk:

```
MEMMAN=NOSWAP
SET PROTSHELL=CMD.EXE
```

The contents of the CONFIG.SYS file on the boot disk, for ISA systems, are:

```
BUFFERS=32
IOPL=YES
MEMMAN=NOSWAP
REM The following line avoids
   an error message but is not
   required:
PROTSHELL=SYSINST1.EXE
SET OS2_SHELL=CMD.EXE
DISKCACHE=64,LW
PROTECTONLY=NO
LIBPATH=.;\;
PAUSEONERROR=NO
CODEPAGE=850
DEVINFO=KBD,US,KEYBOARD.DCP
REM DEVINFO=SCR,EGA,VTB1850.DCP
DEVICE=\DOS.SYS
REM To provide mouse support
   add MOUSE.SYS to the disk and
   unREM the next line:
REM DEVICE=\MOUSE.SYS
SET PATH=.;\
SET DPATH=\;
REM If you do not need to
   print from boot disk, remove
   the following line:
BASEDEV=PRINT01.SYS
BASEDEV=IBM1FLPY.ADD
BASEDEV=IBM1S506.ADD
BASEDEV=IBMINT13.I13
BASEDEV=OS2DASD.DMD
```

```
Directory of A:\*.* Wed 04-29-1992 Files on the boot disk

ANSICALL.DLL        438  ___A_  Copied from Disk 1
BKSCALLS.DLL        401  ___A_  Copied from Disk 1
BMSCALLS.DLL        398  ___A_  Copied from Disk 1
BVGINIT.DLL        9203  ___A_  Copied from Disk 1
BVSVALLS.DLL        454  ___A_  Copied from Disk 1
CLOCK01.SYS        3666  ___A_  Copied from Disk 1
CMD.EXE           87552  ___A_  Copied from Disk 1
CONFIG.SYS          439  ___A_  Copied from Disk 1 and
                                  modified as shown below
COUNTRY.SYS       24604  ___A_  Copied from Disk 1
DOS.SYS            1142  ___A_  Copied from Disk 1
DOSCALL1.DLL      87884  ___A_  Copied from Disk 1
EA_DATA.SF         3584  RHSA_  Created when OS2KRNLI is
                                  copied from Installation
                                  Disk
HARDERR.EXE       14436  ___A_  Copied from Disk 1
IBM1FLPY.ADD      24026  ___A_  Copied from Disk 1
IBM1S506.ADD      12908  ___A_  Copied from Disk 1
IBMINT13.I13       9564  ___A_  Copied from Disk 1
KBD01.SYS          9013  ___A_  Copied from Disk 1
KBDCALLS.DLL        858  ___A_  Copied from Disk 1
KEYBOARD.DCP       5177  ___A_  Copied from Disk 1
MOUCALLS.DLL       1010  ___A_  Copied from Disk 1
MSG.DLL             477  ___A_  Copied from Disk 1
NAMPIPES.DLL        711  ___A_  Copied from Disk 1
NLS.DLL             465  ___A_  Copied from Disk 1
OS2BOOT            1099  RHSA_  Created on boot disk by
                                  SYSINSTX
OS2CHAR.DLL       56320  ___A_  Copied from Disk 1
OS2DASD.DMD       31994  ___A_  Copied from Disk 1
OS2KRNL          716044  ___A_  Copy OS2KRNLI from
                                  Installation Disk to OS2KRNL
OS2LDR            32256  ___A_  Copied from Installation Disk
OS2LDR.MSG         8440  ___A_  Copied from Installation Disk
PRINT01.SYS        8934  ___A_  Copied from Disk 1
                                  (needed only for printing)
QUECALLS.DLL      14994  ___A_  Copied from Disk 1
SCREEN01.SYS       1441  ___A_  Copied from Disk 1
SESMGR.DLL        31256  ___A_  Copied from Disk 1
SYSLEVEL.OS2        169  ___A_  Copied from Disk 1
VIOCALLS.DLL       1825  ___A_  Copied from Disk 1

1,227,406 bytes in 36 file(s)    1,234,944 bytes allocated
222,720 bytes free
```

Figure 6. Files that Enable OS/2 to Be Booted from Floppy Disk

OS/2 Questions and Answers

Doug Azzarito
IBM Corporation
Boca Raton, Florida

Q:

What is OS/2 for Windows? Is it different from OS/2 2.1? Is there any reason to upgrade from OS/2 2.1 to OS/2 for Windows?

A:

OS/2 for Windows has generated some confusion. If you read the fine print on the package, most of the confusion will be answered.

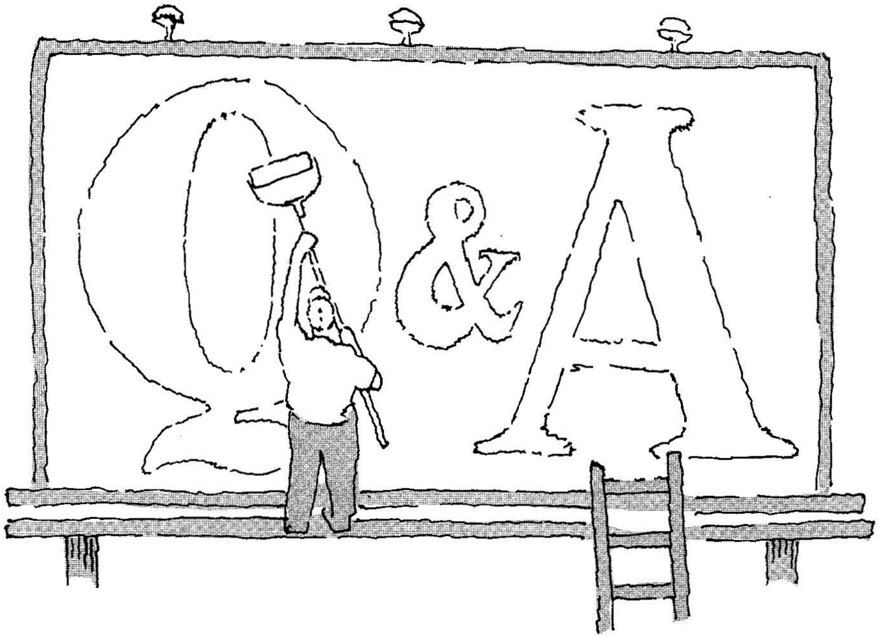
The OS/2 for Windows product is formally called *OS/2 2.1 Special Edition for Use with Windows Version 3.1.* To be sure everyone understands what OS/2 for Windows is, let's go through the technical differences.

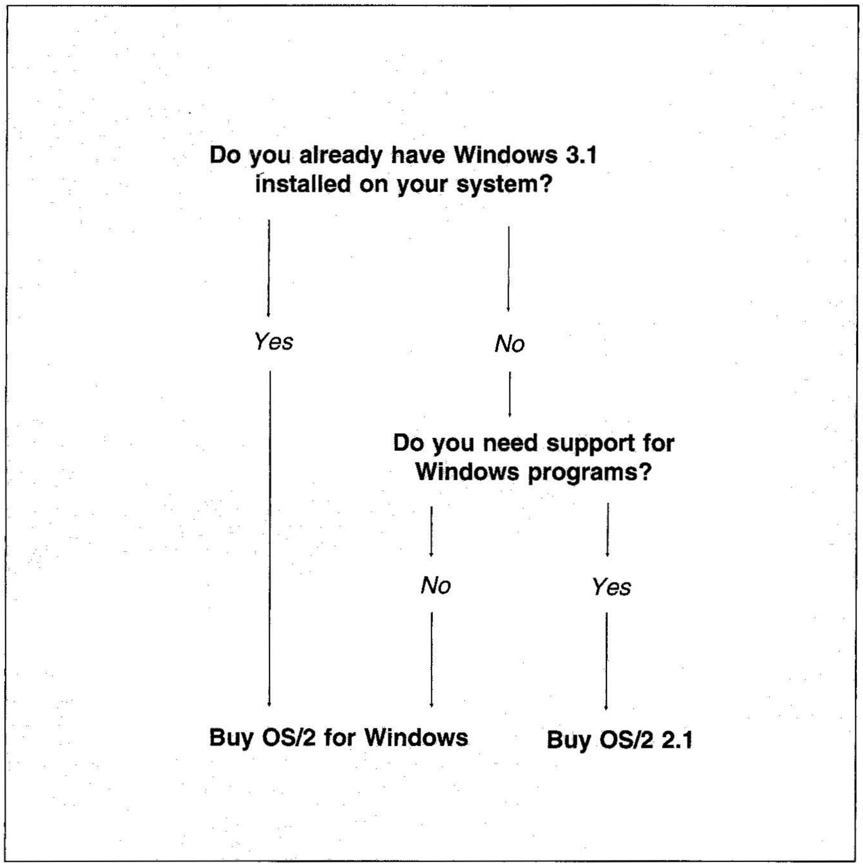
Both OS/2 2.1 and OS/2 for Windows are complete operating systems. The only difference is that OS/2 2.1 includes the full Windows 3.1 package, whereas OS/2 for Windows does not include Windows. But, fret not – if you already have Windows 3.1 on your system, OS/2 for Windows will use your existing Windows 3.1 to create the WIN-OS/2 capability in OS/2.

If you are trying to decide which version of OS/2 to buy, use the simple decision tree to the right.

Let's look at the technical differences in each area of OS/2, so you'll know *exactly* what you're getting.

Installation: OS/2 2.1 comes with WIN-OS/2, and gives you the op-




Do you already have Windows 3.1 installed on your system?
Yes
No
Do you need support for Windows programs?
No
Yes
Buy OS/2 for Windows
Buy OS/2 2.1

tion of installing it. WIN-OS/2 is a complete copy of Windows 3.1, modified to use OS/2's memory management. If you already have a copy of Windows 3.1 on your system, it is left intact, but WIN-OS/2 can read the Windows 3.1 configuration and match it (if you chose this option during OS/2 2.1 installation).

OS/2 for Windows, during installation, looks for an existing copy of Windows 3.1. (*Note:* This works *only* for Windows 3.1! It does *not* work for 3.0, 3.11, Windows for Workgroups, etc.)

If Windows 3.1 is found, OS/2 for Windows modifies Windows 3.1 so that it works with OS/2's memory management (but Windows still works under DOS after the modification).

If Windows 3.1 is not found, OS/2 for Windows installs *without* support for Windows applications. This is a bonus for users of OS/2-only environments: Why pay for a copy of WIN-OS/2 when you don't need it? If you don't need Windows support, OS/2 for Windows is for you!

Running Windows Applications: OS/2 2.1 maintains separate Windows code and configuration files. Changes you make while running Windows under DOS (in either a dual-boot or Boot Manager configuration) will not be made to WIN-OS/2. However, an option during installation does allow changes made to WIN-OS/2 to be copied to your Windows setup. Having two copies of the Windows code and configuration files not only uses extra disk space (Windows 3.1 and WIN-OS/2 each consume close to 10 MB of disk space), but it can be confusing if you switch from DOS/Windows to WIN-OS/2 frequently. OS/2 for Windows uses the same code and configuration files for both DOS/Windows and WIN-OS/2. That saves disk space and configuration confusion. The only benefit you may lose is the performance "tweaking" made to WIN-OS/2 that comes in the OS/2 2.1 package – the IBM programmers made additional performance enhancements to WIN-OS/2 that may make it run slightly faster than native Windows 3.1.

OS/2 2.1 maintains separate Windows code and configuration files.

High-Performance File System: OS/2 2.1 supports HPFS for both the boot drive and any data drives. Even the WIN-OS/2 files can be installed on an HPFS drive.

Since OS/2 for Windows requires Windows 3.1 to be installed *before* installing OS/2, your Windows files must be on a FAT drive. OS/2 can be installed on a separate HPFS drive, and other data drives can be formatted with HPFS.

Other Features: OS/2 for Windows does not contain any new features or fixes, so if you already have OS/2 2.1, there is no need to go to OS/2 for Windows. (If you do, the OS/2 for Windows installation stops once it detects OS/2 2.1.) The only addition to OS/2 for Windows was the inclusion of the S3 video drivers. These drivers were not ready at the time OS/2 2.1 was released, but they have since become available for download from OS/2 bulletin-board systems.

Q:

If I have a CD-ROM drive that is not supported by OS/2, how can I get it to work?

A:

Device support in OS/2 is improving dramatically. The recent Device Driver conference, release of the Device-Driver Source Kit, and additional support from IBM has made it easy for computer device manufacturers to add OS/2 support. If you have a CD-ROM drive, SVGA adapter, SCSI controller or other device, you have two things to consider: Where to get support for your device, and how to add it to OS/2.

Finding information about your device is usually easy. Its manufacturer is the first place to look. If the manufacturer is still not on the OS/2 bandwagon, another call from a customer – you – will help them realize the OS/2 market is growing and cannot be ignored. You shouldn't accept the excuse "We don't know how to develop OS/2 device drivers;" IBM has made developing drivers so easy that you just might find another OS/2 user who has written the driver and released it to the public. For this reason, public OS/2 bulletin boards should be the next place you look. Sources such as CompuServe's OS/2 forum, Internet, or an OS/2 BBS will have a large collection of drivers that you can add to an OS/2 system. (The list of drivers available on the IBM Personal Computer Company BBS appears elsewhere in this magazine.)

Once you find a driver, your next hurdle is to get it added to OS/2. How you do this depends on how the driver comes to you. Sometimes, you will get complete driver installation disks. At other times, you may get only a single driver file, which may mean you'll have to install the driver manually. Here are some examples:

- You purchased the OS/2 2.1 CD version, and you have a Creative Labs SoundBlaster Pro CD-ROM drive. OS/2 2.1 doesn't recognize the drive, so you can't install OS/2. But an OS/2 driver is available for this drive, and a quick call to Creative Labs (or a BBS) puts the driver in your hand.

  Since you want the CD-ROM drive to be recognized during installation, you have to modify the installation disks. The driver file is called SBCD2.ADD. You will copy this file to OS/2 Diskette 1. Then edit the CONFIG.SYS file on this diskette, and add the line:

  ```
  BASEDEV=SBCD2.ADD /P:220
  ```

  This statement loads the device driver during installation, which in turn allows OS/2 to install from the SoundBlaster Pro CD-ROM (the /P:220 is the default port address of the SoundBlaster Pro adapter). Once you make this change, OS/2 installation should be able to read your CD, and proceed with installing OS/2.

- You already have OS/2 2.1 installed, and now you want to add support for the CD-ROM drive (again, we'll use the SoundBlaster Pro CD-ROM as an example). If you received the SBCD2.ADD file on a disk that has the companion files for the device-driver installation program, your job is easy. Open the System Setup folder, select Device Driver Install, and select the drive that contains the driver diskette. This copies the device driver to the proper directory (note that all BASEDEV drivers must be in the \OS2 directory of your boot drive), and it adds the proper statement to CONFIG.SYS. If you have to add the CD-ROM driver manually, copy SBCD2.ADD to the \OS2 directory, then add the following line to your CONFIG.SYS:

  ```
  BASEDEV=SBCD2.ADD /P:220
  ```

  The next step is to install all of the support files needed for the driver. In this case, we need a CD-ROM device manager (CDROM.DMD), a virtual device driver for DOS compatibility (VCDROM.SYS), and a CD file-system driver (CDFS.IFS). All this can be accomplished by using OS/2's Selective Install feature (also in the System Setup folder).

  Start Selective Install, select the CD-ROM Device Support checkmark, and select OK. Select Other from the list of CD-ROM drives, then select OK. On the installation screen, select Install. OS/2 then finds the required files, copies them to the \OS2 subdirectory of your boot drive, and adds the following lines to your CONFIG.SYS:

  ```
  DEVICE=C:\OS2\OS2CDROM.DMD /Q
  DEVICE=C:\OS2\MDOS\VCDROM.SYS
  IFS=C:\OS2\CDFS.IFS /Q
  ```

  The /Q parameter tells the driver and IFS to install in quiet mode. If you want to see messages as they are installed, remove the /Q. If you have trouble with your CD-ROM after installation, make sure these statements (and the files they reference) are where they should be.

Getting all your devices working under OS/2 still takes a bit of work, but the situation is getting better every day. We will probably never see the day when every device is supported on the OS/2 installation diskettes – after all, how many CD-ROMs, SVGA cards, or other devices are supported on the DOS installation diskettes? However, getting OS/2 support disks when you buy a new device is happening more and more often, and should soon be automatic.

***Doug Azzarito** is an advisory programmer on the OS/2 Change Team. He has worked on OS/2 development projects since 1986. Doug is co-author of RBBS-PC, the industry-standard bulletin board software for personal computers. He received a BS in computer science from the University of Florida in 1982.*

Team OS/2 - A Groundswell of Support for OS/2

Dave Whittle
IBM Corporation
Austin, Texas

Reprinted with permission from IBM Personal Systems Technical Solutions *magazine, January/February 1994 issue, pages 17-19.*

You may have heard of Team OS/2, but you might not fully understand what it's all about. Don't feel bad – I started it, and I still don't think I fully understand the phenomenon. I'm certain I don't know everything about every Team OS/2 activity. Literally thousands of enthusiastic volunteers are now part of this "happening." I do know, however, that Team OS/2 has been fueled by the creativity and imagination of many thousands of OS/2 enthusiasts in their pursuit of quality, synergy, and positive relationships. That's worth trying to understand, and I think you'll find it's also worth getting involved.

The Beginning

Team OS/2 has been around, in spirit at least, from the time OS/2 was first conceived by teams of IBM and Microsoft visionaries and programmers looking to replace DOS with a far more capable operating system. It wasn't until February 12, 1992 that it took a recognizable form, when I created TEAMOS2 FORUM on IBM's internal bulletin board. I dedicated the forum to "the discussion of those things that empowered IBMers, working as a team, can do to promote the success of OS/2. The focus here is, through teamwork, creating synergy and combining talents to achieve results far greater than the sum of individual efforts."

The only requirement for membership has been that an individual "make a personal sacrifice, however small, to help others recognize that OS/2 can be the foundation for the next generation of personal computing." At the time Team OS/2 began, OS/2 2.0 was available as beta code in a limited release, enabling a lot of people to experience some of the features that have since made OS/2 such a hit:

- Multitasking that really works
- The powerful but easy Workplace Shell user interface
- The ability to run more PC applications than any operating system or environment in the industry

OS/2 users knew that OS/2 was the underdog in what many perceived as a "war" between OS/2 and DOS/Windows, even though anyone who bought OS/2 got DOS and Windows as well. These users wanted to share their love of OS/2 with others, and that's how Team OS/2 got started.

The Concept

Since the beginning, Team OS/2 has gone wherever Team members have taken it, and has become whatever Team members want it to be. Throughout the world, there are thousands of Team members from a wide variety of OS/2 user communities – both within and outside of IBM. Many of us have found that using OS/2 and computer communications networks has helped us make friends we might otherwise not have made. It has also given us an opportunity to actually put into practice such ideals and principles as a respect for others and a willingness to help others. We don't expect anything in return beyond the intrinsic satisfaction that comes from sharing what we value.

Team OS/2 volunteers have done some amazing things and have a lot to show for their enthusiasm:

- Organizing user-group demonstrations
- Adopting software stores (explaining OS/2 to dealers and sales personnel)
- Setting up booths at fairs
- Demonstrating OS/2 to college professors and classes
- Organizing roving OS/2 help squads to assist vendors in booths at COMDEX**, PC EXPO, and other trade shows
- Working with PRODIGY** and IBM to improve the presence of OS/2 on PRODIGY
- Setting up a Team OS/2 echo on FidoNet
- Writing shareware or other application software for OS/2
- Negotiating the terms under which IBM employees can

release their personally developed OS/2 software for general use

- Helping members of the media understand OS/2
- Getting together with others who use OS/2 to trade tips and experiences
- Starting, supporting, and joining OS/2 user groups and special-interest groups
- Participating in and running OS/2 bulletin boards and online conferences
- Demonstrating OS/2 to new users and encouraging others to try OS/2
- Writing letters to magazines to correct misunderstandings

There have been some exciting times and great moments for Team OS/2. At the first Team OS/2 party at COMDEX in April 1992, the key developers of OS/2 got together with independent software vendors (ISVs), OS/2 customers, marketing personnel, and others to share the excitement of the long-awaited release of the 32-bit OS/2. IBM executive John Soyring, an inspiration to many Team OS/2 members, said it was the first reception he had ever attended that gave him goose bumps. The Chicago jazz band members were so impressed by what they saw happening that they stood in line with everyone else to get their Team OS/2 and "ibm/2" T-shirts.

The T-shirt was inspired by TEAMOS2 FORUM participants who asked for a T-shirt they could wear to identify themselves as empowered members of Team OS/2. The "ibm/2" logo suggests a "new IBM" that respects "the little guy" as well as individual empowerment and initiative. The "/2" emphasizes the ties between OS/2 and this new IBM.

The Commitment

Today, Team OS/2 is open to anyone who wants to be a part of all of this, whether you work for IBM or not. IBM Personal Software Products executives (who also claim membership in Team OS/2) have agreed to support Team OS/2 activities, including occasional Team OS/2 recognition receptions (usually at Fall COMDEX). IBM has a department to respond to requests for assistance from Team OS/2 members, and to support these grassroots marketing efforts, which have been such a key part of OS/2's success.

Team members are familiar with the delightful presence of Vicci Conway and Janet Gobeille, two members of IBM's grassroots department, on the electronic forums and at Team OS/2 hospitality suites at trade shows and conferences. Many of the customers featured in this issue's "Point of View" article (in *IBM Personal Systems Technical Solutions* magazine) are enthusiastic members of Team OS/2.

IBM recognizes that all association with Team OS/2 is purely voluntary, and that there are no mutual expectations or future dependencies. IBM and other companies or individuals with an economic interest in OS/2 are part of Team OS/2 under the same terms as all members – with no strings attached, and with complete respect for the freedom of others and their right to choose their level of commitment and participation.

At the foundation of Team OS/2 are the concepts of quality, imagination, respect, relationships, and teamwork. We don't bash DOS or Windows or other companies or individuals. We understand and appreciate the uniqueness of each individual. We don't take ourselves or OS/2 so seriously that we become fanatics. And, finally, we try to maintain a sense of humor and balance about what we do.

If you choose to become a Team OS/2 member, your participation can take whatever form you choose, consistent with the above concepts. You are free to use the words "Team OS/2" to let others know you are part of this worldwide team. When you say you are a part of Team OS/2, you signal to others that you are willing to help them understand and use OS/2 better. As a Team OS/2 member, you agree not to detract from or dilute the name Team OS/2 by using it in conjunction with activities that disparage or embarrass others.

Thanks for your interest and participation. Here's to a bright future with OS/2, you, and Team OS/2!

Becoming a Team OS/2 Member

To let others know you are part of Team OS/2, and to have your name included in the list we maintain, contact one of the following:

- CompuServe: Vicci Conway at 76711,1123
- Internet: teamos2@vnet.ibm.com
- FidoNet: Janet Gobeille at 1:109/347.3479
- IBMMAIL: USIB45RN at IBMMAIL
- Fax: Team OS/2 Support at 1-512-823-3252

Please include your name, mailing address, phone number, E-mail address, and a one-line description of your ties to and interest in OS/2. (Your mailing address and phone number will not be published in any distribution list.) Please include your experiences with OS/2 and your successes in sharing OS/2 with others, plus anything else you want to share relating to your OS/2 "qualifications."

We will put your name, city, state, E-mail address (of whatever system you include in your application), and description in the public Team OS/2 list, available on the electronic bulletin boards. Your address and phone number will be added to our Team OS/2 database and used only for any necessary future contact, such as Team OS/2 mailings.

***Dave Whittle**, located in Austin, Texas, not only represents IBM Personal Software Products (PSP) on the networks and bulletin boards, but also represents the interests of those on the networks and bulletin boards to PSP. He is the author of* PS/2 Reference Tables *and co-author of* Dvorak's Guide to OS/2 Version 2.1. *He has a BS in accounting and an MBA, both from Brigham Young University.*

Trademarks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation:

IBM, AIX, CICS, DB2/2, Micro Channel, MVS, Operating System/2, OS/2, OS/400, PS/2, Presentation Manager, SOMobjects, ThinkPad, Trackpoint II, VisualAGE, WIN-OS/2, Workplace OS, Workplace Shell, 8516 Touch Display.

The following terms, denoted by a double asterisk (**), are trademarks of the companies noted:

Apple and Macintosh – Apple Computer, Inc.
ATI – ATI Technologies, Inc.
Borland – Borland International, Inc.
C++ – AT&T, Inc.
ChipChat – Cawthon Software Group
COMDEX – Interface Group, Inc.
CompuServe – CompuServe, Inc.
Headland – Headland Technology, Inc.
Hitachi – Hitachi, Ltd.
Intel and Pentium – Intel Corporation
Logitech, Trackman and Mouseman – Logitech Corporation
Micro Focus – Micro Focus Limited
Microsoft and Windows – Microsoft Corporation
Mitsumi – Mitsumi Denki Kabushki Kaisha
NEC – NEC Corporation
Novell, IPX and NetWare – Novell, Inc.
OSF – Open Software Foundation
Oracle – Oracle Corporation
Orchid and Pro Designer – Orchid Technology, Inc.
Panasonic – Matsushita Electric Industrial Co., Ltd.
Pioneer – Pioneer Electronic Corporation
PRODIGY – Prodigy Services Company
Smalltalk and Smalltalk/V – Digitalk, Inc.
Sony – Sony Corporation
Sound Blaster – Creative Labs, Inc.
Taligent – Taligent, Inc.
Toshiba – Toshiba Corporation
Tseng – Tseng Laboratories, Inc.
UNIX – Unix System Laboratories, Inc.
WordPerfect – WordPerfect Corporation
XEROX – Xerox Corporation
X Window System – Massachusetts Institute of Technology

All other product names may be trademarks of their respective companies.

"**Object technology provides a powerful new vision of programming.** (page 11)

"**OpenDoc is document-centered programming.** (page 13)

"**OS/2 2.1 adds CD-ROM support for a range of CD-Technology, Hitachi, NEC, Panasonic, Sony, Texel, and Toshiba CD-ROM drives.** (page 22)

"**There are important differences between OS/2 2.0 and OS/2 2.1 in the installation process for pointing devices.** (page 38)

"**Security in OS/2 is crucial to organizations that must provide enterprise-wide security for their information resources and assets.** (page 42)

"**If the hard drive has 4 MB cylinders, the bootable partition can be as large as 2 GB under FAT, and 4 GB under HPFS.** (page 47)

"**The OS2BOOT file is created, not copied, because it is specific to the computer and to the partition where OS/2 is installed.** (page 52)

"**OS/2 2.1 maintains separate Windows code and configuration files.** (page 56)

"**Team OS/2 volunteers have done some amazing things and have a lot to show for their enthusiasm.** (page 58)